Progress in neural parton distributions

Juan Rojo Chacón
LPTHE - Université Paris VI et Paris VII

Deep Inelastic Scattering 2007 Workshop, April 18th 2007.

The NNPDF Collaboration:
Luigi Del Debbio, Stefano Forte, José I. Latorre,
Andrea Piccione and Juan Rojo,
(2007: +) Richard D. Ball, Alberto Guffanti and Maria Ubiali.
Resuls based on:

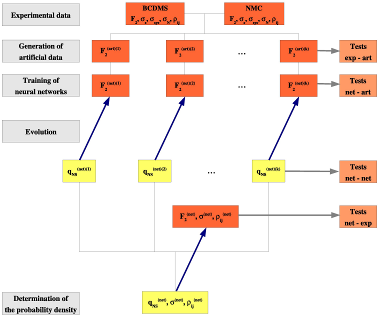
1. JHEP **02** (2002) 062 [arXiv:hep-ph/0204232].
2. JHEP **05** (2005) 080 [arXiv:hep-ph/0501067].
3. *Neural network determination of parton distributions: the nonsinglet case*, **JHEP 07 (2007) 039 [arXiv:hep-ph/0701127]**.

The neural Monte Carlo approach

Basic Idea: Monte Carlo sampling coupled to Neural Network interpolation

- Generate a set of Monte Carlo replicas $\sigma^{(k)}(p_i)$ of the original data set $\sigma^{(\mathrm{data})}(p_i)$, representation of $\mathcal{P}[\sigma(p_i)]$ at discrete set of points p_i
- Train a neural net for each pdf on each replica, obtaining a representation of the pdfs $q_i^{(\mathrm{net})(k)}$
- The set of neural nets is a representation of the probability density:

$$\langle \sigma\left[q_i\right] \rangle = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \sigma\left[q_i^{(\mathrm{net})(k)}\right]$$

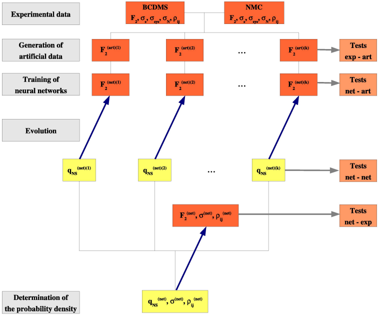


The neural Monte Carlo approach

Basic Idea: Monte Carlo sampling coupled to Neural Network interpolation

- Generate a set of Monte Carlo replicas $\sigma^{(k)}(p_i)$ of the original data set $\sigma^{(\mathrm{data})}(p_i)$, representation of $\mathcal{P}[\sigma(p_i)]$ at discrete set of points p_i
- Train a neural net for each pdf on each replica, obtaining a representation of the pdfs $q_i^{(\mathrm{net})(k)}$
- The set of neural nets is a representation of the probability density:

$$\langle \sigma\left[q_i\right] \rangle = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \sigma\left[q_i^{(\mathrm{net})(k)}\right]$$

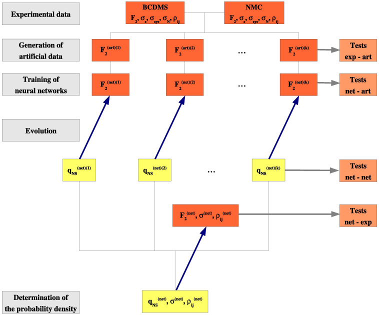


The neural Monte Carlo approach

Basic Idea: Monte Carlo sampling coupled to Neural Network interpolation

- Generate a set of Monte Carlo replicas $\sigma^{(k)}(p_i)$ of the original data set $\sigma^{(\mathrm{data})}(p_i)$, representation of $\mathcal{P}[\sigma(p_i)]$ at discrete set of points p_i
- Train a neural net for each pdf on each replica, obtaining a representation of the pdfs $q_i^{(\mathrm{net})(k)}$
- The set of neural nets is a representation of the probability density:

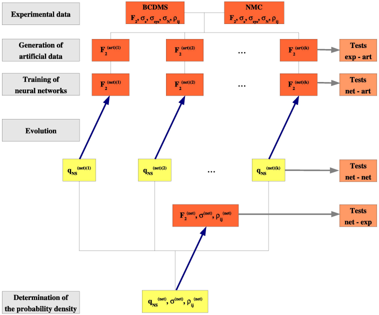
$$\langle \sigma\left[q_i\right] \rangle = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \sigma\left[{q_i}^{(\mathrm{net})(k)} \right]$$



The neural Monte Carlo approach

Basic Idea: Monte Carlo sampling coupled to Neural Network interpolation

- Generate a set of Monte Carlo replicas $\sigma^{(k)}(p_i)$ of the original data set $\sigma^{(\mathrm{data})}(p_i)$, representation of $\mathcal{P}[\sigma(p_i)]$ at discrete set of points p_i
- Train a neural net for each pdf on each replica, obtaining a representation of the pdfs $q_i^{(\mathrm{net})(k)}$
- The set of neural nets is a representation of the probability density:



$$\langle \sigma \left[q_i \right] \rangle = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \sigma \left[q_i{}^{(\mathrm{net})(k)} \right]$$

Neural nets

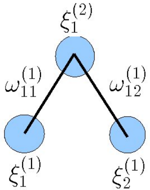
What is a neural net? Just a particular choice of function

$$\xi_i^{(l+1)} = g\left(\sum_{j=1}^{n(l)} \omega_{ij}^{(l)} \xi_j^l\right), \quad g(x) = \frac{1}{1+e^x}, \quad l = 2, \ldots, L$$

where $\omega_{ij}^{(l)}$ are the *weights* and $\xi_i^{(l+1)}$ the *activation state* of each neuron.

- Functional form $q(x) = f\left[x, \{A_i\}\right] = A_1 x^{A_2}(1-x)^{A_3}$
- Neural net $q(x) = f\left[x, \{\omega_{ij}\}\right] = g\left(\sum_{j=1}^{n(L-1)} \omega_{ij}^{(L-1)} \xi_j^{L-1}\right)$, $\xi_1^{(1)} = x$

Simple net: Architecture 2-1 $\rightarrow \xi_1^{(2)} = \left[1 + \exp\left(\omega_{11}^{(1)}\xi_1^{(1)} + \omega_{12}^{(1)}\xi_2^{(1)}\right)\right]^{-1}$



Neural nets

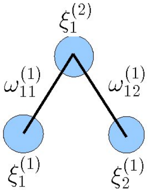
What is a neural net? Just a particular choice of function

$$\xi_i^{(l+1)} = g\left(\sum_{j=1}^{n(l)} \omega_{ij}^{(l)} \xi_j^l\right), \quad g(x) = \frac{1}{1+e^x}, \quad l = 2, \ldots, L$$

where $\omega_{ij}^{(l)}$ are the *weights* and $\xi_i^{(l+1)}$ the *activation state* of each neuron.

- Functional form $q(x) = f\left[x, \{A_i\}\right] = A_1 x^{A_2} (1-x)^{A_3}$
- Neural net $q(x) = f\left[x, \{\omega_{ij}\}\right] = g\left(\sum_{j=1}^{n(L-1)} \omega_{ij}^{(L-1)} \xi_j^{L-1}\right)$, $\xi_1^{(1)} = x$

Simple net: Architecture 2-1 $\rightarrow \xi_1^{(2)} = \left[1 + \exp\left(\omega_{11}^{(1)} \xi_1^{(1)} + \omega_{12}^{(1)} \xi_2^{(1)}\right)\right]^{-1}$



Neural nets

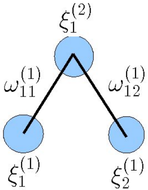
What is a neural net? Just a particular choice of function

$$\xi_i^{(l+1)} = g\left(\sum_{j=1}^{n(l)} \omega_{ij}^{(l)} \xi_j^{l}\right), \quad g(x) = \frac{1}{1+e^x}, \quad l = 2, \ldots, L$$

where $\omega_{ij}^{(l)}$ are the *weights* and $\xi_i^{(l+1)}$ the *activation state* of each neuron.

- Functional form $q(x) = f\left[x, \{A_i\}\right] = A_1 x^{A_2} (1-x)^{A_3}$
- Neural net $q(x) = f\left[x, \{\omega_{ij}\}\right] = g\left(\sum_{j=1}^{n(L-1)} \omega_{ij}^{(L-1)} \xi_j^{L-1}\right)$, $\xi_1^{(1)} = x$

Simple net: Architecture 2-1 $\rightarrow \xi_1^{(2)} = \left[1 + \exp\left(\omega_{11}^{(1)}\xi_1^{(1)} + \omega_{12}^{(1)}\xi_2^{(1)}\right)\right]^{-1}$



Neural nets

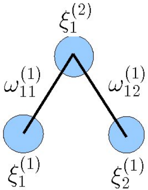
What is a neural net? Just a particular choice of function

$$\xi_i^{(l+1)} = g\left(\sum_{j=1}^{n(l)} \omega_{ij}^{(l)} \xi_j^l\right), \quad g(x) = \frac{1}{1+e^x}, \quad l = 2, \ldots, L$$

where $\omega_{ij}^{(l)}$ are the *weights* and $\xi_i^{(l+1)}$ the *activation state* of each neuron.

- Functional form $q(x) = f\left[x, \{A_i\}\right] = A_1 x^{A_2}(1-x)^{A_3}$
- Neural net $q(x) = f\left[x, \{\omega_{ij}\}\right] = g\left(\sum_{j=1}^{n(L-1)} \omega_{ij}^{(L-1)} \xi_j^{L-1}\right)$, $\xi_1^{(1)} = x$

Simple net: Architecture 2-1 $\rightarrow \xi_1^{(2)} = \left[1 + \exp\left(\omega_{11}^{(1)}\xi_1^{(1)} + \omega_{12}^{(1)}\xi_2^{(1)}\right)\right]^{-1}$



Methodological issues

Methodological issues

Many different ingredients in the neural Monte Carlo approach to parton distributions: artificial data generation, neural network training, genetic minimization, preprocessing, result validation ...

Let us concentrate on a couple of the newest ones:

- Stopping criterion
- Stability estimators

N.B. Unless otherwise specified, results shown belong to hep-ph/0701127

Methodological issues

Many different ingredients in the neural Monte Carlo approach to parton distributions: artificial data generation, neural network training, genetic minimization, preprocessing, result validation ...
Let us concentrate on a couple of the newest ones:

- Stopping criterion
- Stability estimators

N.B. Unless otherwise specified, results shown belong to hep-ph/0701127

Methodological issues

Many different ingredients in the neural Monte Carlo approach to parton distributions: artificial data generation, neural network training, genetic minimization, preprocessing, result validation ...
Let us concentrate on a couple of the newest ones:

- Stopping criterion
- Stability estimators

N.B. Unless otherwise specified, results shown belong to hep-ph/0701127

Methodological issues

Many different ingredients in the neural Monte Carlo approach to parton distributions: artificial data generation, neural network training, genetic minimization, preprocessing, result validation ...
Let us concentrate on a couple of the newest ones:

- Stopping criterion
- Stability estimators

N.B. Unless otherwise specified, results shown belong to hep-ph/0701127

When to stop a fit?

In a standard fit, look for minimum χ^2 for given parametrization. However ...

- If basis too large $\rightarrow$ convergence never reached
- If basis too small $\rightarrow$ parametrization bias

How can one obtain an unbiased compromise? For neural nets, smoothness decreases as fit quality improves...

$\rightarrow$ Stop before fitting statistical noise (overlearning).

When to stop a fit?

In a standard fit, look for minimum χ^2 for given parametrization. However ...

- If basis too large $\rightarrow$ convergence never reached
- If basis too small $\rightarrow$ parametrization bias

How can one obtain an unbiased compromise? For neural nets, smoothness decreases as fit quality improves...

$\rightarrow$ Stop before fitting statistical noise (overlearning).

When to stop a fit?

In a standard fit, look for minimum χ^2 for given parametrization. However ...

- If basis too large $\longrightarrow$ convergence never reached
- If basis too small $\longrightarrow$ parametrization bias

How can one obtain an unbiased compromise? For neural nets, smoothness decreases as fit quality improves...

$\longrightarrow$ Stop before fitting statistical noise (overlearning).

When to stop a fit?

In a standard fit, look for minimum χ^2 for given parametrization. However ...

- If basis too large $\longrightarrow$ convergence never reached
- If basis too small $\longrightarrow$ parametrization bias

How can one obtain an unbiased compromise? For neural nets, smoothness decreases as fit quality improves...

$\longrightarrow$ Stop before fitting statistical noise (overlearning).

When to stop a fit?

In a standard fit, look for minimum χ^2 for given parametrization. However ...

- If basis too large $\longrightarrow$ convergence never reached
- If basis too small $\longrightarrow$ parametrization bias

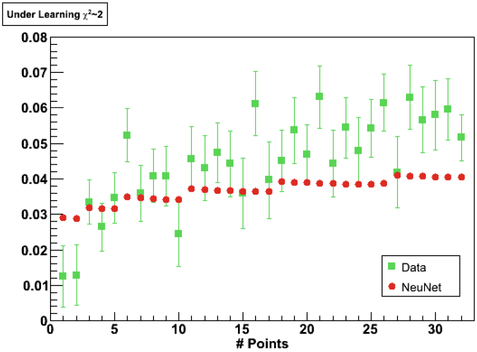
How can one obtain an unbiased compromise? For neural nets, smoothness decreases as fit quality improves...

$\longrightarrow$ Stop before fitting statistical noise (overlearning).

When to stop a fit?

→ Stop before fitting statistical noise (overlearning).
Ex.: fitting a neural net to signal+noise pseudodata.

Underlearning

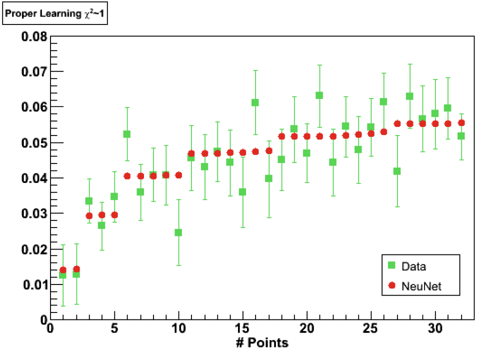


(Gross features of data already captured)

When to stop a fit?

→ Stop before fitting statistical noise (overlearning).

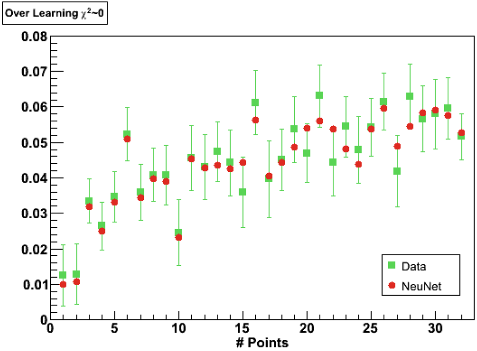
Proper learning



When to stop a fit?

→ Stop before fitting statistical noise (overlearning).

Overlearning



The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice?

At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice?
At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice? At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice?
At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice?
At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

The stopping criterion

So stop the fit before overlearning sets in ... How does this work in practice? At each Genetic Algorithm iteration, χ^2 either decreases or unchanged

1. Divide the data set into training and validation sets
2. Minimize χ^2 of training set, monitor χ^2 of validation set
3. Stop minimization when validation χ^2 begins to rise

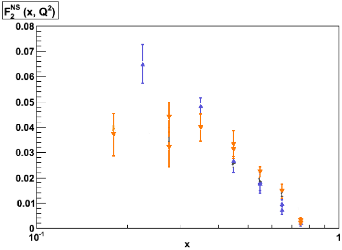
The transition from the underlearning to the overlearning regime monotonic with neural networks.

N.B. This stopping criterion could also be applied using standard polynomials (but impractical, a very large number of parameters required + non monotonic transition to overlearning regime).

An explicit example

Let us see in practice how the overlearning stopping criterion works:

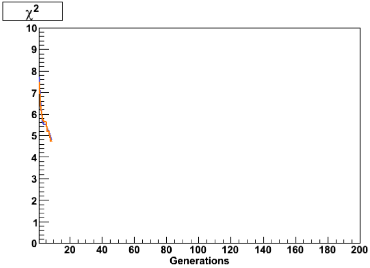
On your marks, get ready ...

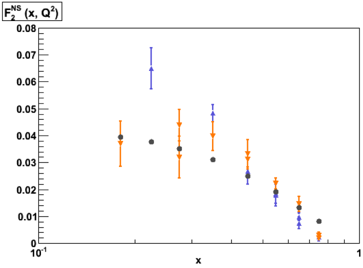


An explicit example

Let us see in practice how the overlearning stopping criterion works: Compare training and validation χ^2.
(*N.B.* With GA, the network fit quality improves monotonically with the number of iterations)

Go!



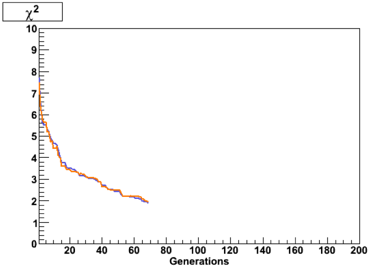


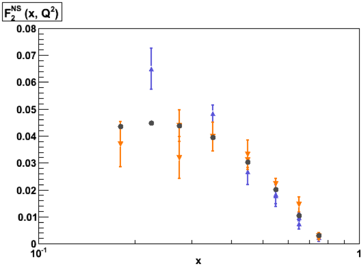
Underlearning, continue the minimization

An explicit example

Let us see in practice how the overlearning stopping criterion works:

Stop!



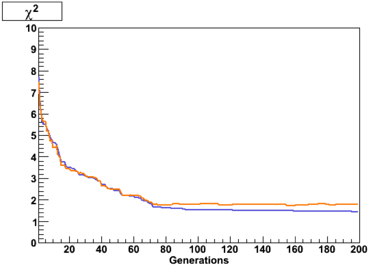


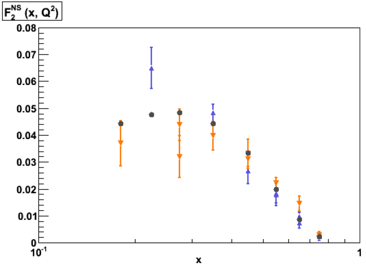
Onset of overlearning, stop the minimization.

An explicit example

Let us see in practice how the overlearning stopping criterion works:

Too late!

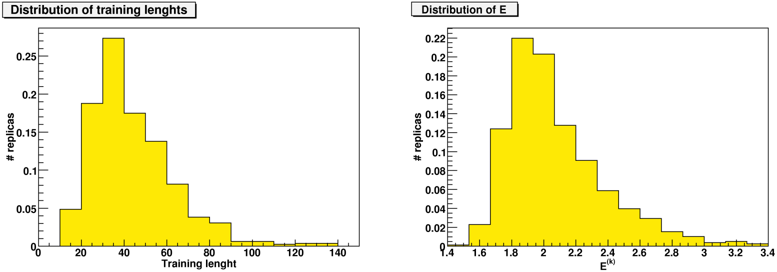




Deep in the overlearning region, fitting statistical noise ...

Does it work?

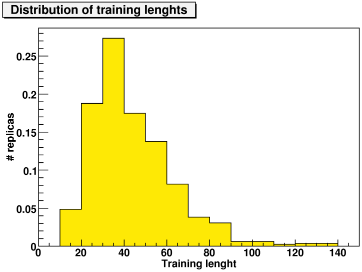
Distribution of χ^2 of pseudo-data and training lenghts

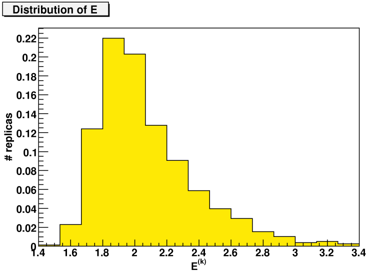


1. Poissonian distribution of training lenghts
2. For best fit, average χ^2 of replicas ~ 2, while when averaging over replicas $\chi^2 \sim 1$.
3. Total training time is optimized (never underlearn nor overlearn) $\longrightarrow$ efficient neural fitting.

Does it work?

Distribution of χ^2 of pseudo-data and training lenghts

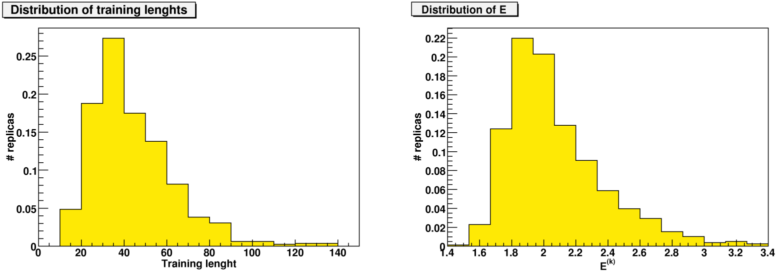




1. Poissonian distribution of training lenghts
2. For best fit, average χ^2 of replicas ~ 2, while when averaging over replicas $\chi^2 \sim 1$.
3. Total training time is optimized (never underlearn nor overlearn) $\longrightarrow$ efficient neural fitting.

Does it work?

Distribution of χ^2 of pseudo-data and training lenghts



1. Poissonian distribution of training lenghts
2. For best fit, average χ^2 of replicas ~ 2, while when averaging over replicas $\chi^2 \sim 1$.
3. Total training time is optimized (never underlearn nor overlearn) $\rightarrow$ efficient neural fitting.

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q]\rangle = \sqrt{\left\langle \frac{\left(\langle q_i\rangle_{(1)} - \langle q_i\rangle_{(2)}\right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\rm dat}}$$

where $\sigma[q_i]$ = error on $\langle q_i\rangle$ = error on $q_i/\sqrt{N_{\rm rep}}$.

Compute $\langle d[q]\rangle$ and $\langle d[\sigma]\rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q]\rangle \sim 1$, $\langle d[\sigma]\rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q]\rangle \sim \sqrt{\frac{N_{\rm rep}}{2}} \left\langle \frac{\left|\langle q_i\rangle_{(1)} - \langle q_i\rangle_{(2)}\right|}{\sigma[q_i]} \right\rangle_{\rm dat}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q] \rangle = \sqrt{ \left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\mathrm{dat}} }$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i / \sqrt{N_{\mathrm{rep}}}$.

Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\mathrm{rep}}}{2}} \left\langle \frac{\left| \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right|}{\sigma[q_i]} \right\rangle_{\mathrm{dat}}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q] \rangle = \sqrt{\left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\rm dat}}$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i/\sqrt{N_{\rm rep}}$.

Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\rm rep}}{2}} \left\langle \frac{\left| \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right|}{\sigma[q_i]} \right\rangle_{\rm dat}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q] \rangle = \sqrt{\left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\rm dat}}$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i/\sqrt{N_{\rm rep}}$.

Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\rm rep}}{2}} \left\langle \frac{\left| \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right|}{\sigma[q_i]} \right\rangle_{\rm dat}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q] \rangle = \sqrt{\left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\rm dat}}$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i/\sqrt{N_{\rm rep}}$.

Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\rm rep}}{2}} \left\langle \frac{|\, \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \,|}{\sigma[q_i]} \right\rangle_{\rm dat}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified
Define RMS distance

$$\langle d[q] \rangle = \sqrt{\left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\mathrm{dat}}}$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i / \sqrt{N_{\mathrm{rep}}}$.
Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.
Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\mathrm{rep}}}{2}} \left\langle \frac{\left| \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right|}{\sigma[q_i]} \right\rangle_{\mathrm{dat}}$$

Stability

Check stability and accuracy of our results (both for central values and for errors) when parameters of fit modified

Define RMS distance

$$\langle d[q] \rangle = \sqrt{ \left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\mathrm{dat}} }$$

where $\sigma[q_i]$ = error on $\langle q_i \rangle$ = error on $q_i/\sqrt{N_{\mathrm{rep}}}$.

Compute $\langle d[q] \rangle$ and $\langle d[\sigma] \rangle$ both in data region and in extrapolation region.

Statistical expectations:

- For statistically equivalent fits (different set of MC replicas, different net architecture) we expect $\langle d[q] \rangle \sim 1$, $\langle d[\sigma] \rangle \sim 1$.
- For statistically nonequivalent fits (different perturbative order, different α_s value) we expect (e.g. for central values)

$$\langle d[q] \rangle \sim \sqrt{\frac{N_{\mathrm{rep}}}{2}} \left\langle \frac{\left| \langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right|}{\sigma[q_i]} \right\rangle_{\mathrm{dat}}$$

Great expectations?

Indeed we observe the expected behavior:

Architecture	2-4-3-1 vs. 2-5-3-1
$\langle d\,[q]\rangle_{\rm dat}$	0.9
$\langle d\,[q]\rangle_{\rm extra}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.4

Perturbative order	LO vs. NLO
$\langle d\,[q]\rangle_{\rm dat}$	10.2
$\langle d\,[q]\rangle_{\rm extra}$	1.2
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	2.2
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.3

Independence of the net architecture
$\rightarrow$ Explicit confirmation of parametrization invariance : stability of both central values and errors!

Great expectations?

Indeed we observe the expected behavior:

Architecture	2-4-3-1 vs. 2-5-3-1
$\langle d\,[q]\rangle_{\rm dat}$	0.9
$\langle d\,[q]\rangle_{\rm extra}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.4

Perturbative order	LO vs. NLO
$\langle d\,[q]\rangle_{\rm dat}$	10.2
$\langle d\,[q]\rangle_{\rm extra}$	1.2
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	2.2
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.3

Independence of the net architecture

$\rightarrow$ Explicit confirmation of parametrization invariance : stability of both central values and errors!

Great expectations?

Indeed we observe the expected behavior:

Architecture	2-4-3-1 vs. 2-5-3-1
$\langle d\,[q] \rangle_{\mathrm{dat}}$	0.9
$\langle d\,[q] \rangle_{\mathrm{extra}}$	0.9
$\langle d\,[\sigma_q] \rangle_{\mathrm{dat}}$	0.9
$\langle d\,[\sigma_q] \rangle_{\mathrm{extra}}$	1.4

Perturbative order	LO vs. NLO
$\langle d\,[q] \rangle_{\mathrm{dat}}$	10.2
$\langle d\,[q] \rangle_{\mathrm{extra}}$	1.2
$\langle d\,[\sigma_q] \rangle_{\mathrm{dat}}$	2.2
$\langle d\,[\sigma_q] \rangle_{\mathrm{extra}}$	1.3

Independence of the net architecture
$\rightarrow$ Explicit confirmation of parametrization invariance : stability of both central values and errors!

Great expectations?

Indeed we observe the expected behavior:

Architecture	2-4-3-1 vs. 2-5-3-1
$\langle d\,[q]\rangle_{\rm dat}$	0.9
$\langle d\,[q]\rangle_{\rm extra}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	0.9
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.4

Perturbative order	LO vs. NLO
$\langle d\,[q]\rangle_{\rm dat}$	10.2
$\langle d\,[q]\rangle_{\rm extra}$	1.2
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	2.2
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.3

Independence of the net architecture
→ Explicit confirmation of parametrization invariance : stability of both central values and errors!

The nonsinglet case

Summary of the analysis

1. Data: $F_2^p(x,Q^2) - F_2^d(x,Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\rm NS}(x,Q_0^2) \equiv \left(u+\bar{u}-(d+\bar{d})\right)(x,Q_0^2)$ at $Q_0^2 = 2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\rm NS}(x,Q^2) = \frac{1}{6}x\int_x^1 \frac{dy}{y}\tilde{\Gamma}(y,\alpha_s\left(Q^2\right),\alpha_s\left(Q_0^2\right))\,q_{\rm NS}\left(\frac{x}{y},Q_0^2\right) \ .$$

Results available from `http://sophia.ecm.ub.es/nnpdf`

See hep-ph/0701127 for all the technical details ...

Summary of the analysis

1. Data: $F_2^p(x,Q^2) - F_2^d(x,Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\rm NS}(x,Q_0^2) \equiv \left(u+\bar{u}-(d+\bar{d})\right)(x,Q_0^2)$ at $Q_0^2 = 2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\rm NS}(x,Q^2) = \frac{1}{6}x\int_x^1 \frac{dy}{y}\tilde{\Gamma}\left(y,\alpha_s\left(Q^2\right),\alpha_s\left(Q_0^2\right)\right)q_{\rm NS}\left(\frac{x}{y},Q_0^2\right) .$$

Results available from `http://sophia.ecm.ub.es/nnpdf`

See hep-ph/0701127 for all the technical details ...

Summary of the analysis

1. Data: $F_2^p(x,Q^2) - F_2^d(x,Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\mathrm{NS}}(x,Q_0^2) \equiv \left(u+\bar{u}-(d+\bar{d})\right)(x,Q_0^2)$ at $Q_0^2 = 2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\mathrm{NS}}(x,Q^2) = \frac{1}{6}x\int_x^1 \frac{dy}{y}\tilde{\Gamma}(y,\alpha_s\left(Q^2\right),\alpha_s\left(Q_0^2\right))\, q_{\mathrm{NS}}\left(\frac{x}{y},Q_0^2\right) \ .$$

Results available from http://sophia.ecm.ub.es/nnpdf

See hep-ph/0701127 for all the technical details ...

Summary of the analysis

1. Data: $F_2^p(x,Q^2) - F_2^d(x,Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\rm NS}(x,Q_0^2) \equiv \left(u+\bar{u}-(d+\bar{d})\right)(x,Q_0^2)$ at $Q_0^2=2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\rm NS}(x,Q^2) = \frac{1}{6}x\int_x^1 \frac{dy}{y}\tilde{\Gamma}(y,\alpha_s\left(Q^2\right),\alpha_s\left(Q_0^2\right))\, q_{\rm NS}\left(\frac{x}{y},Q_0^2\right) \; .$$

Results available from http://sophia.ecm.ub.es/nnpdf

See hep-ph/0701127 for all the technical details ...

Summary of the analysis

1. Data: $F_2^p(x,Q^2) - F_2^d(x,Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\mathrm{NS}}(x,Q_0^2) \equiv \left(u+\bar{u}-(d+\bar{d})\right)(x,Q_0^2)$ at $Q_0^2 = 2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\mathrm{NS}}(x,Q^2) = \frac{1}{6} x \int_x^1 \frac{dy}{y} \tilde{\Gamma}(y, \alpha_s\left(Q^2\right), \alpha_s\left(Q_0^2\right)) \, q_{\mathrm{NS}}\left(\frac{x}{y}, Q_0^2\right) \ .$$

Results available from `http://sophia.ecm.ub.es/nnpdf`

See hep-ph/0701127 for all the technical details ...

Summary of the analysis

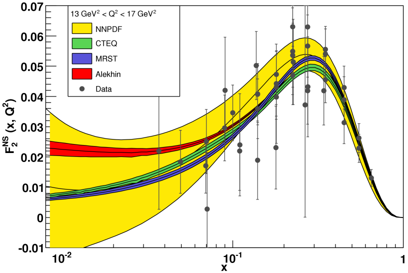
1. Data: $F_2^p(x, Q^2) - F_2^d(x, Q^2)$ from NMC and BCDMS (483 points with $Q^2 \geq 3$ GeV^2)
2. Determination of $q^{\rm NS}(x, Q_0^2) \equiv \left(u + \bar{u} - (d + \bar{d})\right)(x, Q_0^2)$ at $Q_0^2 = 2$ GeV^2 at LO, NLO, NNLO, and for different values of $\alpha_s(M_Z^2)$.
3. New fast and efficient implementation of NNLO parton evolution (mixed x/N space), benchmarked with the LH tables (Salam/Vogt, 2006).

$$F_2^{\rm NS}(x, Q^2) = \frac{1}{6} x \int_x^1 \frac{dy}{y} \tilde{\Gamma}(y, \alpha_s\left(Q^2\right), \alpha_s\left(Q_0^2\right)) q_{\rm NS}\left(\frac{x}{y}, Q_0^2\right) .$$

Results available from `http://sophia.ecm.ub.es/nnpdf`

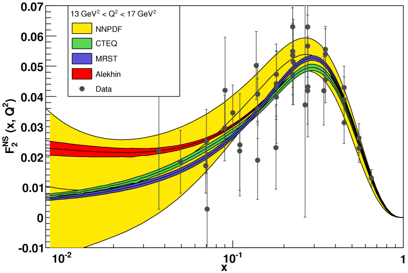
See hep-ph/0701127 for all the technical details ...

Comparison to other approaches



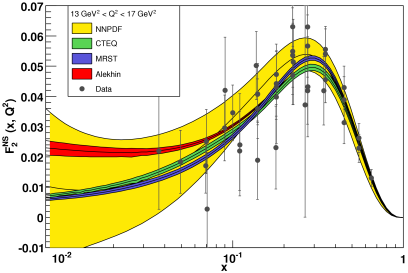
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region: absence of functional form bias.
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



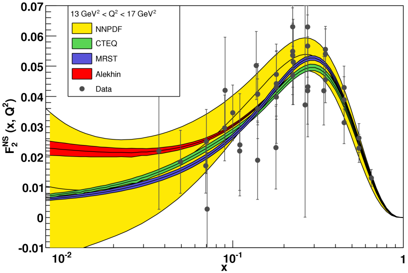
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region: absence of functional form bias.
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



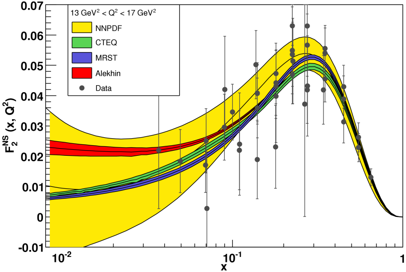
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region: absence of functional form bias.
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



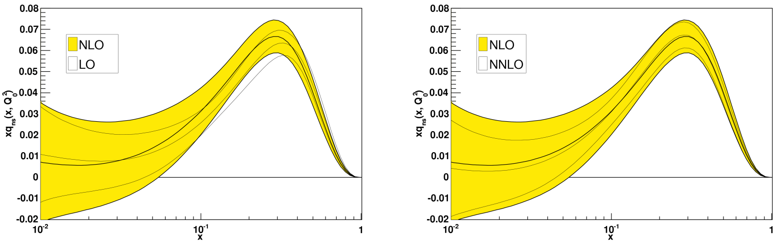
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region: absence of functional form bias.
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



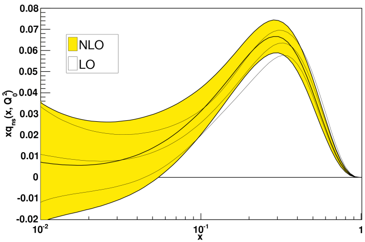
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region: absence of functional form bias.
3. Clear effect of error increase in extrapolation region.

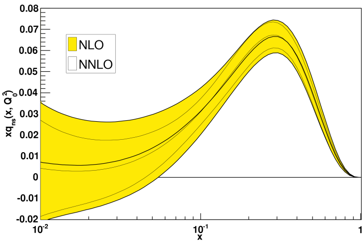
Theoretical uncertainties



1. Same quality of the fit ($\chi^2/N_{\rm dat} \sim 0.75$) at LO, NLO, NNLO
2. NNLO terms negligible within errors
3. LO/NLO differ within 3 σ: NLO terms absorbed in BC.

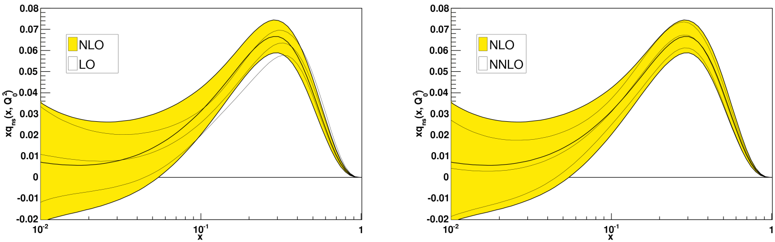
Theoretical uncertainties





1. Same quality of the fit ($\chi^2/N_{\rm dat} \sim 0.75$) at LO, NLO, NNLO
2. NNLO terms negligible within errors
3. LO/NLO differ within 3 σ: NLO terms absorbed in BC.

Theoretical uncertainties



1. Same quality of the fit ($\chi^2/N_{\rm dat} \sim 0.75$) at LO, NLO, NNLO
2. NNLO terms negligible within errors
3. LO/NLO differ within 3 σ: NLO terms absorbed in BC.

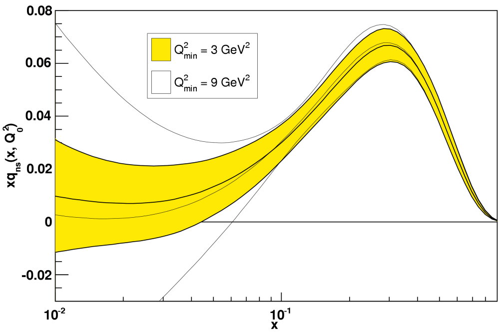
Determination of $\alpha_s(M_Z^2)$ - I

We do not fit $\alpha_s(M_Z^2)$ together with $q_{\rm NS}(x, Q_0^2)$ but take it from world average: $\alpha_s(M_Z^2)_{\rm NLO} = 0.118 \pm 0.002$.

$\alpha_s(M_Z^2)$	0.116	0.118	0.120
χ^2	0.743	0.750	0.744
$\langle d\,[q]\rangle_{\rm dat}$	3.8	-	4.1
$\langle d\,[q]\rangle_{\rm extra}$	0.8	-	0.7
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	1.6	-	2.4
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.4	-	1.5

Fit results suggest that $\alpha_s(M_Z^2)$ from NS data has larger uncertainties than those of world average.

Kinematical cuts



Uncertainty in extrapolation region (small x) increases when kinematical cut in Q^2 is raised (as it should be!).

The nonsinglet case: Applications (preliminary)

Determination of $\Delta\chi^2$

Many hints that some data sets used in global fits inconsistent $\rightarrow$ Some uncertainties underestimated $\rightarrow$ The $\Delta\chi^2$ which corresponds to $1-\sigma$ *errors on PDFs* no longer textbook value $\Delta\chi^2 = 1$.

In our approach $\Delta\chi^2$ can be quantitatively determined $\rightarrow$ The result will tell whether inconsistent data are present.
Compute variance σ_{χ^2} of the χ^2 when fit repeated many times.
For reference fit (preliminary result):

$$\Delta\chi^2 \equiv \sqrt{N_{\rm rep}}\sigma_{\chi^2} \sim 1.7$$

The result implies that most Non Singlet world data form a consistent data set, but some subset inconsistent (subset of NMC data).

Determination of $\Delta\chi^2$

Many hints that some data sets used in global fits inconsistent $\rightarrow$ Some uncertainties underestimated $\rightarrow$ The $\Delta\chi^2$ which corresponds to $1-\sigma$ *errors on PDFs* no longer textbook value $\Delta\chi^2 = 1$.

In our approach $\Delta\chi^2$ can be quantitatively determined $\rightarrow$ The result will tell whether inconsistent data are present.

Compute variance σ_{χ^2} of the χ^2 when fit repeated many times.

For reference fit (preliminary result):

$$\Delta\chi^2 \equiv \sqrt{N_{\mathrm{rep}}}\sigma_{\chi^2} \sim 1.7$$

The result implies that most Non Singlet world data form a consistent data set, but some subset inconsistent (subset of NMC data).

Determination of $\Delta\chi^2$

Many hints that some data sets used in global fits inconsistent $\rightarrow$ Some uncertainties underestimated $\rightarrow$ The $\Delta\chi^2$ which corresponds to $1-\sigma$ *errors on PDFs* no longer textbook value $\Delta\chi^2 = 1$.

In our approach $\Delta\chi^2$ can be quantitatively determined $\rightarrow$ The result will tell whether inconsistent data are present.
Compute variance σ_{χ^2} of the χ^2 when fit repeated many times.
For reference fit (preliminary result):

$$\Delta\chi^2 \equiv \sqrt{N_{\mathrm{rep}}}\sigma_{\chi^2} \sim 1.7$$

The result implies that most Non Singlet world data form a consistent data set, but some subset inconsistent (subset of NMC data).

Determination of $\Delta\chi^2$

Many hints that some data sets used in global fits inconsistent $\rightarrow$ Some uncertainties underestimated $\rightarrow$ The $\Delta\chi^2$ which corresponds to $1-\sigma$ *errors on PDFs* no longer textbook value $\Delta\chi^2 = 1$.

In our approach $\Delta\chi^2$ can be quantitatively determined $\rightarrow$ The result will tell whether inconsistent data are present.
Compute variance σ_{χ^2} of the χ^2 when fit repeated many times.
For reference fit (preliminary result):

$$\Delta\chi^2 \equiv \sqrt{N_{\rm rep}}\sigma_{\chi^2} \sim 1.7$$

The result implies that most Non Singlet world data form a consistent data set, but some subset inconsistent (subset of NMC data).

Determination of $\Delta\chi^2$

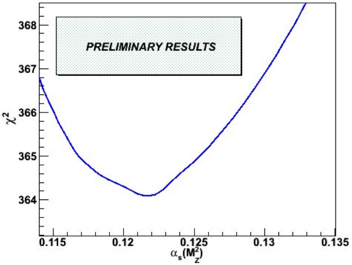
Many hints that some data sets used in global fits inconsistent $\rightarrow$ Some uncertainties underestimated $\rightarrow$ The $\Delta\chi^2$ which corresponds to $1-\sigma$ *errors on PDFs* no longer textbook value $\Delta\chi^2 = 1$.

In our approach $\Delta\chi^2$ can be quantitatively determined $\rightarrow$ The result will tell whether inconsistent data are present.
Compute variance σ_{χ^2} of the χ^2 when fit repeated many times.
For reference fit (preliminary result):

$$\Delta\chi^2 \equiv \sqrt{N_{\rm rep}}\sigma_{\chi^2} \sim 1.7$$

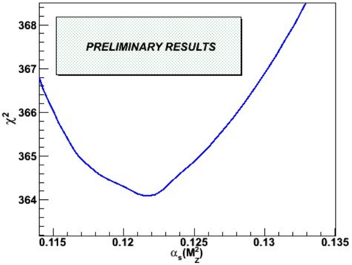
The result implies that most Non Singlet world data form a consistent data set, but some subset inconsistent (subset of NMC data).

Determination of $\alpha_s(M_Z^2)$ - II



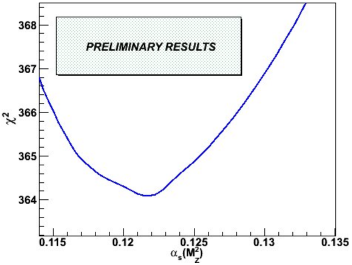
$\alpha_s(M_Z^2)$ can be determined from NS data $\rightarrow$ uncertainty larger than 0.002.
Due to lack of param. bias? Compare with parametrization-independent determination from NS truncated moments $\rightarrow$
$\alpha_s(M_Z^2)_{\mathrm{NLO}} = 0.124 + 0.004 - 0.007$ (exp.), S. Forte et al., hep-ph/0205286.

Determination of $\alpha_s(M_Z^2)$ - II



$\alpha_s(M_Z^2)$ can be determined from NS data $\rightarrow$ uncertainty larger than 0.002.
Due to lack of param. bias? Compare with parametrization-independent determination from NS truncated moments $\rightarrow$
$\alpha_s(M_Z^2)_{\rm NLO} = 0.124 + 0.004 - 0.007$ (exp.), S. Forte et al., hep-ph/0205286.

Determination of $\alpha_s(M_Z^2)$ - II



$\alpha_s(M_Z^2)$ can be determined from NS data $\rightarrow$ uncertainty larger than 0.002.
Due to lack of param. bias? Compare with parametrization-independent determination from NS truncated moments $\rightarrow$
$\alpha_s(M_Z^2)_{\rm NLO} = 0.124 + 0.004 - 0.007$ (exp.), S. Forte et al., hep-ph/0205286.

Status of the singlet case

The singlet case

- Data sets: SLAC, NMC, BCDMS structure function $F_2(x, Q^2)$ and HERA reduced cross sections $\widetilde{\sigma}^{NC}(x, Q^2)$ and $\widetilde{\sigma}^{CC}(x, Q^2)$.
- Fitted parton distributions (neural nets):

$$\Sigma(x, Q_0^2) \; , \quad q_{\mathrm{NS}}(x, Q_0^2) \; , \quad g(x, Q_0^2)$$

- ZM-VFN treatment of heavy flavors.
- Dynamical stopping criterion, weighted Genetic Algorithms minimization.

The singlet case

- Data sets: SLAC, NMC, BCDMS structure function $F_2(x, Q^2)$ and HERA reduced cross sections $\widetilde{\sigma}^{NC}(x, Q^2)$ and $\widetilde{\sigma}^{CC}(x, Q^2)$.
- Fitted parton distributions (neural nets):

$$\Sigma(x, Q_0^2)\ , \quad q_{\rm NS}(x, Q_0^2)\ , \quad g(x, Q_0^2)$$

- ZM-VFN treatment of heavy flavors.
- Dynamical stopping criterion, weighted Genetic Algorithms minimization.

The singlet case

- Data sets: SLAC, NMC, BCDMS structure function $F_2(x, Q^2)$ and HERA reduced cross sections $\widetilde{\sigma}^{NC}(x, Q^2)$ and $\widetilde{\sigma}^{CC}(x, Q^2)$.
- Fitted parton distributions (neural nets):

$$\Sigma(x, Q_0^2)\ , \quad q_{\rm NS}(x, Q_0^2)\ , \quad g(x, Q_0^2)$$

- ZM-VFN treatment of heavy flavors.
- Dynamical stopping criterion, weighted Genetic Algorithms minimization.

The singlet case

- Data sets: SLAC, NMC, BCDMS structure function $F_2(x, Q^2)$ and HERA reduced cross sections $\widetilde{\sigma}^{NC}(x, Q^2)$ and $\widetilde{\sigma}^{CC}(x, Q^2)$.
- Fitted parton distributions (neural nets):

$$\Sigma(x, Q_0^2) \ , \quad q_{\mathrm{NS}}(x, Q_0^2) \ , \quad g(x, Q_0^2)$$

- ZM-VFN treatment of heavy flavors.
- Dynamical stopping criterion, weighted Genetic Algorithms minimization.

Project status

- Increased manpower of the NNPDF collaboration (RDB, AG, MU)
- October 2006 - February 2007: Nonsinglet code extended to the singlet sector (data generation, evolution, neural parton fitting, validation)
- Expected timescale for full NLO DIS fit: Summer 2007.
- Expected timescale for full NLO global neural parton fit: Before end 2007.

Project status

- Increased manpower of the NNPDF collaboration (RDB, AG, MU)
- October 2006 - February 2007: Nonsinglet code extended to the singlet sector (data generation, evolution, neural parton fitting, validation)
- Expected timescale for full NLO DIS fit: Summer 2007.
- Expected timescale for full NLO global neural parton fit: Before end 2007.

Project status

- Increased manpower of the NNPDF collaboration (RDB, AG, MU)
- October 2006 - February 2007: Nonsinglet code extended to the singlet sector (data generation, evolution, neural parton fitting, validation)
- Expected timescale for full NLO DIS fit: Summer 2007.
- Expected timescale for full NLO global neural parton fit: Before end 2007.

Project status

- Increased manpower of the NNPDF collaboration (RDB, AG, MU)
- October 2006 - February 2007: Nonsinglet code extended to the singlet sector (data generation, evolution, neural parton fitting, validation)
- Expected timescale for full NLO DIS fit: Summer 2007.
- Expected timescale for full NLO global neural parton fit: Before end 2007.

Conclusions and outlook

Conclusions and outlook

- Deeper understanding of neural parton fitting achieved (overlearning stopping criterion, stability estimators ...)
- Non-singlet parton fitting completed.
- Be prepared for first neural parton set by summer 2007!

Conclusions and outlook

- Deeper understanding of neural parton fitting achieved (overlearning stopping criterion, stability estimators ...)
- Non-singlet parton fitting completed.
- Be prepared for first neural parton set by summer 2007!

Conclusions and outlook

- Deeper understanding of neural parton fitting achieved (overlearning stopping criterion, stability estimators ...)
- Non-singlet parton fitting completed.
- Be prepared for first neural parton set by summer 2007!

EXTRA SLIDES

Statistical estimators

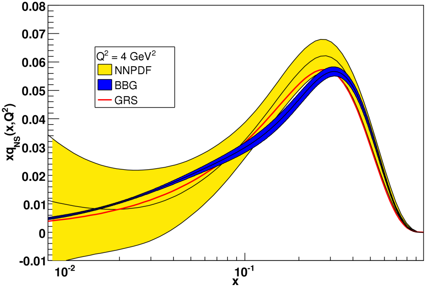
	Total	NMC	BCDMS
χ^2_{tot}	0.75	0.72	0.78
$\langle E \rangle$	2.27	1.99	2.52
$r\left[F_2^{\mathrm{NS}}\right]$	0.81	0.66	0.95
$\left\langle \sigma^{(\mathrm{exp})} \right\rangle_{\mathrm{dat}}$	0.011	0.017	0.006
$\left\langle \sigma^{(\mathrm{net})} \right\rangle_{\mathrm{dat}}$	0.006	0.009	0.004
$r\left[\sigma^{(\mathrm{net})}\right]$	0.59	-0.04	0.86
$\left\langle \rho^{(\mathrm{exp})} \right\rangle_{\mathrm{dat}}$	0.11	0.39	0.16
$\left\langle \rho^{(\mathrm{net})} \right\rangle_{\mathrm{dat}}$	0.46	0.42	0.50
$r\left[\rho^{(\mathrm{net})}\right]$	0.15	0.25	0.04
$\left\langle \mathrm{cov}^{(\mathrm{exp})} \right\rangle_{\mathrm{dat}}$	$8.6\ 10^{-6}$	$1.0\ 10^{-5}$	$7.2\ 10^{-6}$
$\left\langle \mathrm{cov}^{(\mathrm{net})} \right\rangle_{\mathrm{dat}}$	$2.1\ 10^{-5}$	$3.8\ 10^{-5}$	$6.9\ 10^{-6}$
$r\left[\mathrm{cov}^{(\mathrm{net})}\right]$	0.24	0.23	0.57

Higher Twist

No evidence for Higher Twist found in experimental data:

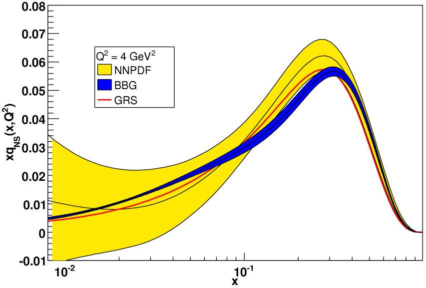
Fit	$Q^2_{\rm min} = 3\ \mathrm{GeV}^2$+HT	$Q^2_{\rm min} = 5\ \mathrm{GeV}^2$	$Q^2_{\rm min} = 5\ \mathrm{GeV}^2$+HT
χ^2	0.76	0.79	0.78
$\langle d\,[q]\rangle_{\rm dat}$	2.9	0.8	3.2
$\langle d\,[q]\rangle_{\rm extra}$	1.4	0.8	0.9
$\langle d\,[\sigma_q]\rangle_{\rm dat}$	1.2	1.5	1.9
$\langle d\,[\sigma_q]\rangle_{\rm extra}$	1.3	1.8	2.3

Comparison to other approaches



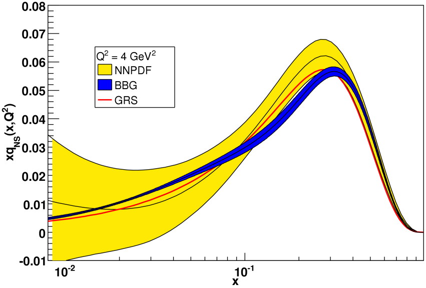
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



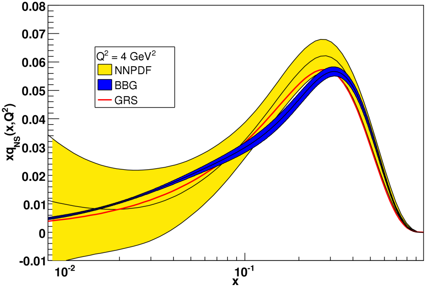
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



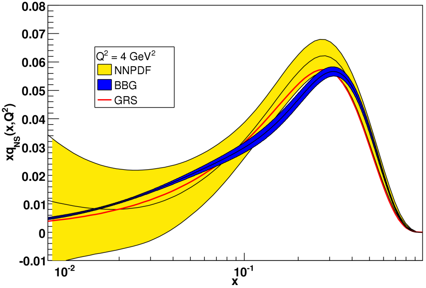
1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region
3. Clear effect of error increase in extrapolation region.

Comparison to other approaches



1. Underestimated uncertainties in existing fits (both global and NS fits)
2. Larger uncertainties both in data and in extrapolation region
3. Clear effect of error increase in extrapolation region.