Application of different fitting algorithms to the estimation of neutrino mass.

A. Nozik

The problem

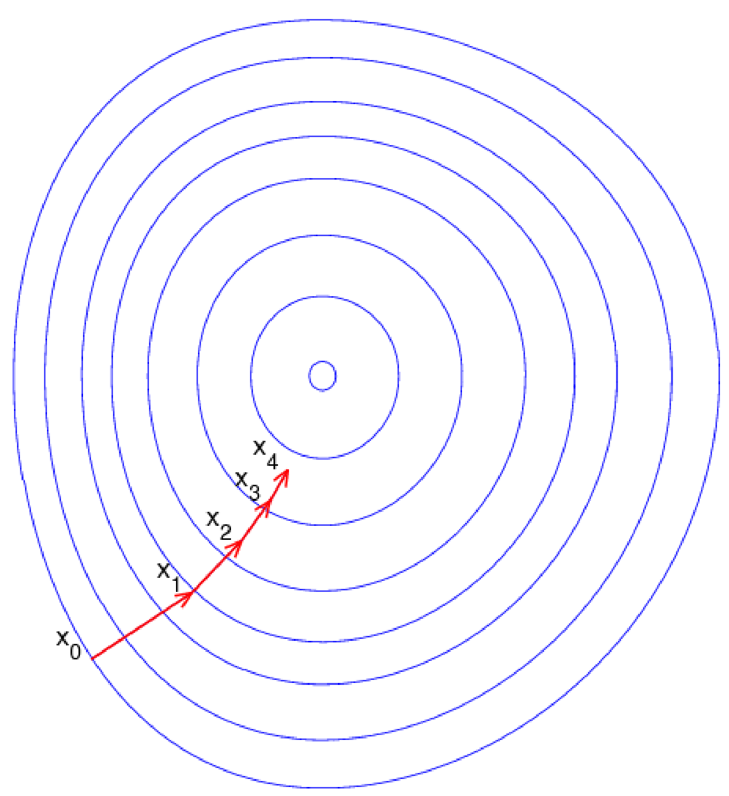
- Four-parameter fit: squared neutrino mass, spectrum endpoint, normalizing factor and background level.
- Very high correlation between parameters (up to 90%).
- Relatively long spectrum calculation time (at least triple integral for each parameter set).

Fitting algorithms used

- Gradient descent methods (MINUIT MIGRAD, FUMILIE).
- Simplex minimization or maximization (MINUIT SIMPLEX).
- Stright-ahead function shape analysis (MINUIT MINOS).
- Tkachov's quasi-optimal weights.

Gradient descent

- Iterative procedure.
- The direction of each step is ∇F
- The length of the step is determined from second order derivatives.
- Iteration ends when ∇F is close to zero (more complex criteria could be used).



Градиентные методики

$$\bar{x}_{k+1} = \bar{x}_k - \lambda_k \cdot \frac{\nabla f(x_k)}{|\nabla f(x_k)|}$$

$$\lambda_k = \frac{|\nabla f(x_k)|^3}{\nabla f^T \cdot \nabla^2 f \cdot \nabla f}$$

Метод Ньютона:

$$\bar{x}_{k+1} = \bar{x}_k - [\nabla^2 f(x_k)]^{-1} \cdot \nabla f(x_k) = \bar{x}_k - H_k \cdot \nabla f(x_k)$$

Pros and cons of gradient descent

- The best choice for Gaussian distributions without strong correlations (relatively fast and reliable).
- Implemented in some well developed packages.
- Requires first and sometimes second derivatives.
- Requires explicit or implicit Hessian matrix inversion (which in case of strong correlations leads to fit crash).

I found (maybe it is an artifact of Java implementation) that errors generated by MIGRAD during fit in some cases are totally wrong, while MINOS errors are fine.

Simplex (Nelder-Mead) optimization

A modern version of direct maximum (minimum) search.
Uses shrinking polyhedrons in space of parameters.

- Does not require derivatives at all (nearly indifferent to correlations).
- Requires tremendous number of function calls.
- Does not have good internal goodness of fit criterion.

Quasi-optimal weights

The method of quasi-optimal weights is developed by F. Tkachov and is based on well known generalized method of moments.

If φ is a function of experimental values X then one can construct the mean value of φ in two different ways:

$$\langle \phi \rangle_{theor} = E[\phi(X)]$$ - expected value of φ

$$\langle \phi \rangle_{\exp} = \frac{1}{N} \sum_i \phi(X_i)$$ - experimental mean of φ

Since expected value depends on parameter value θ, one can write an equation:

$$\langle \phi \rangle_{theor}(\theta) = \langle \phi \rangle_{\exp}(X_i)$$

Estimations made by this procedure are automatically unbiased and consistent but not necessarily efficient. Efficiency could be acquired by letting φ to depend on "real" parameter value θ (witch is unknown). In this case φ(θ, X) is optimal weight. A quasi-optimal weight could be obtained by calculating φ in point θ_0 which is close to the real value θ.

КОВ на практике

Оптимальный вес: $\varphi_{opt} = \frac{\partial L(X|\theta_{true})}{\partial \theta}$

Истинное значение

Квазиоптимальный вес: $\theta_{true} \rightarrow \theta_0$

$$L = \prod_i \frac{1}{\sqrt{(2\pi)}\sigma} \exp\left(\frac{-(S_i(\theta) - X_i)^2}{2\sigma^2}\right)$$

С одной стороны, знание распределения величины – дополнительная информация. С другой, эта информация неявным образом все равно используется в градиентных методах.

КОВ на практике

$$Eq_k = \sum_i \frac{S_i(\theta) - X_i}{\sigma_i^2(\theta_0)} \cdot \frac{\partial S_i(\theta)}{\partial \theta_k}\bigg|_{\theta_0} = 0$$

$$\frac{\partial Eq_k}{\partial \theta_l} = \sum_i \frac{1}{\sigma_i^2(\theta_0)} \cdot \frac{\partial S_i(\theta)}{\partial \theta_l} \cdot \frac{\partial S_i(\theta)}{\partial \theta_k}\bigg|_{\theta_0}$$

Решение:

$$\theta_{(p+1)} = \theta_{(p)} - \left[\frac{\partial Eq_k}{\partial \theta_l}\right]_{\theta_{(p)}}^{-1} \cdot Eq\left(\theta_{(p)}\right)$$

или

$$\theta_{(p+1)} = \theta_{(p)} - \left[\frac{\partial Eq_k}{\partial \theta_l}\right]_{\theta_{(0)}}^{-1} \cdot Eq\left(\theta_{(p)}\right)$$

Pros and cons of QOW

- Much faster than gradient methods (requires less function calls).
- Needs first derivatives, but does not require second ones (no Hessian inversion).
- In case starting point is far from maximum, needs few runs in order to find the optimal weight.

Comparison of different implementations

Some algorithms were implemented in the JAVA program (the program is the single framework where different methods could be applied to the data in any order).

- Gradient descent using JMINUIT (http://java.freehep.org/freehep-jminuit/).
- Simplex maximization form Michael Thomas Flanagan's Java Scientific Library (http://www.ee.ucl.ac.uk/~mflanaga/java/)
- Quasi-optimal weights (my own implementation).

Comparison table

Number of function and function derivative calls to fit neutrino mass.
A derivative calculation is in general much more time consuming than function calculation.

	Function calls	Derivative calls
Simplex	>300	0
Gradient	>30	>30
QOW	~10	~10*

* - could be lowered to 1 if the alternative procedure is used (in this case number of function calls could be about 15).

Conclusions:

• In case time does not matter, it is better to use simplex algorithms.
• For fast calculations QOW proved itself best.
• In case of strong parameter correlation gradient methods such as MIGRAD should be used carefully. For MINUIT only MNHESSE or MINOS errors are reliable