Description of the S-programs for calculating the MLE of k-monotone density

by Fadoua Balabdaoui and Jon A. Wellner

December 7, 2004

- *Armijo*: in order to ensure monotonicity of the algorithm, this function looks for a $\lambda \in (0,1)$ such that $\bar{f} + \lambda(f_q - \bar{f})$ has a larger likelihood than that of $\bar{f}$, where $\bar{f}$ is the estimator obtained "outside the quadratic approximation", whereas f_q, as the notation suggests, is the minimizer of the quadratic approximation of the loglikelihood function.

- *minusloglik*: returns -loglikelihood at a vertex $\theta > X_{(n)}$:

$$k \log(\theta) - \frac{1}{n} \sum_{i=1}^{n} (\theta - X_i)^{k-1} .$$

  The latter is minimized at $\theta^{(0)}$ which is taken as an initial candidate for a support point.

- *EvaluateMatf*: given a set of vertices $\underline{\theta} = (\theta_1, \cdots, \theta_m)$, the function returns the $n \times m$ matrix $\left(g_{\theta_j}(X_i)\right)_{i,j}$, $1 \le i \le n, 1 \le j \le m$ where

$$g_\theta(x) = \frac{k(\theta - x)_+^{k-1}}{\theta^k}.$$

- *LoglikFunc*: calculates $-(1/n)\times$ loglikelihood + the penalty term at a current iterate f:

$$-\frac{1}{n} \sum_{i=1} n \log(f(X_i)) + \int_0^\infty f(t)dt.$$

- *IndexFuncMLE*: finds the index of the support point to be eliminated from the vector of all support points of the current iterate (reduction step).

- *CalculateOptMLE*: solves a linear system (see Balabadoui and Wellner 2004, Part2) in order to find the minimizer of -loglikelihood over the cone generated by the vertices of the current iterate.

- *DirecDerMLE*: calculates the directional derivative of the quadratic approximation of -loglikelihood at some point θ.

- *FindMinimMLE*: finds the minimum of the directional derivative over a chosen fine grid. The minimizer will be added to the current set of support points.

- *SuppReducAlgoMLE*: this is the main function that calls all the other sub-programs, and it calculates after a finite number of iterations an approximation of the MLE of a k-monotone density based on n independent observations. The function takes the following arguments:

    1. The integer k (≥ 2): the smoothness parameter.
    2. X: the vector of observations.
    3. *prec*: a parameter that controls how much fine the user wants to choose the grid over which the minimization problem is solved.
    4. *eps*: the tolerance chosen by the user (should be very small). If the next iteration results in a directional derivative that is bigger or equal than $-eps$, then the algorithm stops.

The algorithm returns two vectors called S and C: S is the vector of support points and C is the corresponding vector of weights. The user might use any other software to plot the MLE in the direct and inverse problems.