Shifting sands

A man with a hammer

Home | About Me

Friday, December 26, 2014

Fitting a mixture of independent Poisson distributions

This is an example from Zucchini & MacDonald's book on Hidden Markov Models for Time Series (exercise 1.3).

The data is annual counts of earthquakes of magnitude 7 or greater, which exhibits both overdispersion for a Poisson (where the mean should equal the variance) as well as serial dependence.

The aim is to fit a mixture of m independent Poisson distributions to this data, using the non-linear minimizer nlm to minimize the negative log-likelihood of the Poisson mixture.

Sounds easy right? I have not done much optimisation stuff so it took me longer than it probably should have. There's little better for the ego then getting stuck on things by page 10.

Constrained to unconstrained

There are two sets of parameters, the lambda parameters to the Poisson distributions and the delta mixing distribution, the latter giving the proportions of each distribution to mix.

The Poisson parameters must be greater than zero and there are m of them. The mixing distribution values must all sum to one, and the first mixing value is implied by the m-1 subsequent values.

These are the constraints (lambdas greater than zero, deltas sum to one), and there are 2m - 1 values (m lambdas, m-1 deltas). We need to transform these to unconstrained values for use with nlm (becoming eta and tau respectively).

The formulas for transformation are reproduced from the book here

such as **nlm**. One possibility is to maximize the likelihood with respect to the $2m-1$ unconstrained parameters

$$\eta_i = \log \lambda_i \quad (i = 1, \ldots, m)$$

and

$$\tau_i = \log \left(\frac{\delta_i}{1 - \sum_{j=2}^{m} \delta_j} \right) \quad (i = 2, \ldots, m).$$

One recovers the original parameters via

$$\lambda_i = e^{\eta_i}, \quad (i = 1, \ldots, m),$$

$$\delta_i = \frac{e^{\tau_i}}{1 + \sum_{j=2}^{m} e^{\tau_j}} \quad (i = 2, \ldots, m),$$

and $\delta_1 = 1 - \sum_{j=2}^{m} \delta_i$. See Exercise 3.

These transformed values are combined in to one parameter vector and passed to nlm. We also pass a function that calculates the negative log likelihood.

In the function, we first convert the parameters from the unconstrained "working" form to the constrained "natural" form using the inverse transform, and then use these values to do the likelihood calculation.

You can see the code here.

Outro

Now that I look over the code and write this out it does seem all fairly straightforward and relatively trivial, but it was new to me and took a while, so I figured other people might

Popular Posts

- Intro to Curl Noise
- Using apply, sapply, lapply in R
- Streaming OANDA with python and ZeroMQ
- Equity Ranking Backtest with Python/Pandas
- Apple is really cool

Blog Roll

- Quant Mashup | Quantocracy
- Abnormal Returns
- R-bloggers
- Eran Raviv
- R by examples
- Quant News

Subscribe To

- Posts
- Comments

Blog Archive

- ► 2016 (1)
- ► 2015 (6)
- ▼ 2014 (12)
  - ▼ December (3)
    - Fitting a mixture of independent Poisson distribut...
    - Does Trend Following Work?
    - Book Review: R Object-oriented Programming
  - ► September (1)
  - ► July (1)
  - ► June (1)
  - ► May (4)
  - ► April (1)
  - ► January (1)
- ► 2013 (17)
- ► 2012 (12)
- ► 2011 (9)

find the example useful as well.

The code has three examples for the cases m = 2, 3, and 4. Obviously feel free to mess around with the initial parameter estimates to see what comes up.

It does seem to match the fitted models from the book, which I have copied here:

Table 1.2 *Poisson independent mixture models fitted to the earthquakes series. The number of components is m, the mixing probabilities are denoted by δ_i, and the component means by λ_i. The maximized likelihood is L.*

model	i	δ_i	λ_i	$-\log L$	mean	variance
$m = 1$	1	1.000	19.364	391.9189	19.364	19.364
$m = 2$	1	0.676	15.777	360.3690	19.364	46.182
	2	0.324	26.840			
$m = 3$	1	0.278	12.736	356.8489	19.364	51.170
	2	0.593	19.785			
	3	0.130	31.629			
$m = 4$	1	0.093	10.584	356.7337	19.364	51.638
	2	0.354	15.528			
	3	0.437	20.969			
	4	0.116	32.079			
observations					19.364	51.573

For all the details, check out the book, chapter 1!

Posted by Pete at 9:39 PM

Labels: github, Hidden Markov Models, HMM, HMMs, optimisation, R, r code, time series, zucchini

No comments:

Post a Comment

Enter Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)