2. Log-likelihood gradient

The *energy* of a Markov network is defined as

$$E(y;\alpha,\beta) = -\sum_i \alpha_i y_i - \sum_{i \neq j} \beta_{ij} y_i y_j.$$

The energy function can be used to define the log-likelihood of a given y vector as

$$\log \mathcal{L}(y;\alpha,\beta) = -E(y;\alpha,\beta) - \log(Z(\alpha,\beta)).$$

Here, $Z(\alpha,\beta)$ is the *partition function*, a scaling factor defined as

$$Z(\alpha,\beta) = \sum_{y \in Y} e^{-E(y;\alpha,\beta)},$$

where Y is the set of all possible y vectors.

The partial derivative of the log-likelihood with respect to α_i is

$$\frac{\partial}{\partial \alpha_i} \log \mathcal{L}(y;\alpha,\beta) = y_i - p(y_i;\alpha,\beta).$$

The first term, y_i, is zero if species i is absent in the observed assemblage and one if it's present. The latter term, $p(y_i;\alpha,\beta)$, describes the expected probability of observing species i under the current values of α and β. It comes from the derivative of the partition function, as derived in (learning Boltzmann machines, Murphy 2012, etc.). Following the gradient of α_i adjusts the expected probability of observing species i until it matches the observed value and the two terms in the gradient cancel one another out.

The partial derivative of the log-likelihood with respect to β_{ij} can be derived similarly as

$$\frac{\partial}{\partial \alpha_i} \log \mathcal{L}(y;\alpha,\beta) = y_i y_j - p(y_i y_j;\alpha,\beta).$$

Following this gradient adjusts the expected probability of co-occurrence between species i and species j until this value matches the observed co-occurrence frequency and the two terms in the gradient cancel one another out.

References