Algorithm for step size of IIS based CRF optimizer

简述：

在 CRF 中，基于 IIS 的 Algorithm S 算法需要求对于全局所有 x_seq 最大的 step size 倒数:

$$max[\sum_c \sum_k f_k(y_{seq_c}, x_{seq}) + \sum_c \sum_k g_k(y_{seq_c}, x_{seq})];$$

由于对于某个 x_seq 来说，y_seq 需要枚举，基本上普通的枚举方法是 ***intractable*** 的，因此需要使用 Dynamic Programming 的方法来求这个最大值。

由于都是在枚举的情况中寻求使得 P 最大的序列的值，因此可以从 Viterbi Algorithm 中修改以适合当前任务。

构想：

我们最终要求的是 $y_{seq}' = arg\,max_{y_{seq}}[\sum_c \sum_k f_k(y_{seq_c}, x_{seq}) + \sum_c \sum_k g_k(y_{seq_c}, x_{seq})];$

同样地，定义一个前向向量 α 代表由前一步转到当前一步，出现 y 标签的 potential，定义如下：

$$\alpha_i(x_{seq}) = \alpha_{i-1}(x_{seq}) \times M(x_{seq})$$

其中的 M(x_seq)与原来的 M(x_seq)有所不同，原来的有考虑参数进去，而这里只是求最大的和，不是 inference，无需使用到模型的参数，因此参数都设为 1.

$$M(y_{front} = y', y_{rare} = y | x_{seq}) = \sum_k f_k(y_{front} = y', y_{rare} = y, x_{seq}) + \sum_k g_k(y_{rare} = y, x_{seq});$$

而原来的 M 是这样的:

$$\begin{aligned} M_i(y', y \mid \mathbf{x}) &= \exp(\Lambda_i(y', y \mid \mathbf{x})) \\ \Lambda_i(y', y \mid \mathbf{x}) &= \sum_k \lambda_k f_k(e_i, \mathbf{Y}|_{e_i} = (y', y), \mathbf{x}) + \\ &\quad \sum_k \mu_k g_k(v_i, \mathbf{Y}|_{v_i} = y, \mathbf{x}), \end{aligned}$$

剩下的跟 Viterbi 一样了，求出最大可能的序列，然后这里还多了一步，还要反过来再求这条序列的 active feature 数目。

实现：

基本上可以用