Technical Notes: Models of Multiscale Agglomerative Settlement

Phil Chodrow

May 23, 2017

These are technical notes for specifying and learning models of urban growth. See the overview notes for motivations and interpretations. The two algorithms depend on the following notations:

1. G is the set of possible sites, which are indexed by k.
2. Each site possesses a time-dependent indicator variable $W_k(t)$ indicating whether site k is occupied.
3. The set of occupied sites is $\mathcal{W}(t) = \{k \in G \mid W_k(t) = 1\}$. The set of unoccupied sites is $\bar{W}(t) = G \setminus \mathcal{W}(t)$.
4. The sites are related by a metric $d_{k\ell}$ between sites k and ℓ. The distance matrix D is given by $D = [d_{k\ell}]$.

1 Two Supervised Models

We will first elaborate two supervised models for studying settlement, both of which treat type-conditional settlement as a monotonic transformation of a distance-weighted proximity function. We define

$$p_{k|r}(\theta) = \mathbb{P}(W_k(t+1) = 1 | R_k(t+1) = r; \theta) \tag{1}$$

and

$$q_{kr}(\theta) = \frac{p_{k|r}(\theta)}{\sum_{\rho \in \mathcal{R}} p_{k|\rho}(\theta)}. \tag{2}$$

The likelihood of settlement at site k is then

$$\mathbb{P}(W_k(t+1) = 1) = \sum_{r \in \mathcal{R}} p_{k|r} q_{kr}. \tag{3}$$

Models may be specified by specifying the functional form of $p_{k|r}(\theta)$. Two possibilities are:

$$p_{k|r}(\theta) = \alpha_r \frac{\sum_{j \in \mathcal{W}_r(t)} d_{ij}^{-\gamma_r}}{\sum_{j \in G} d_{ij}^{-\gamma_u}} + \beta_r \tag{4}$$

$$p_{k|r}(\theta) = \alpha_r \sigma \left(\sum_{j \in \mathcal{W}_r(t)} d_{ij}^{-\gamma_r} \right) + \beta_r \ , \tag{5}$$

where $\mathcal{W}_r(t) = \{k \in \mathcal{W}(t) \mid R_k = r\}$. In each expression, α_r is interpretable as the relative rate of growth of settlements of type r, β_r is the background rate at which settlements appear without any local interactions, and γ_r is a spatial dispersion parameter which is large when settlements are tightly concentrated in space and small when they are highly disperse. The two specifications differ only in how they transform the distance-weighting sum $\sum_{j \in \mathcal{W}_r(t)} d_{ij}^{-\gamma_r}$ into an appropriately normalized probability.

We observe the data D consisting of measures $x_k \in \{0,1\}$ for each $k \in \bar{\mathcal{W}}(t)$. Suppose that, in addition, we hypothetically observed the latent types of potential settlements $R_k \in \{0,1\}$, with $R_k = 0$ for rural settlement and $R_k = 1$ for urban settlement. Then, we could write the complete data likelihood in the form

$$\mathcal{L}(X, Z; \theta) = \prod_{k \in \bar{\mathcal{W}}(t), r \in \mathcal{R}} \left[q_{kr}(\theta) p_{k|r}(\theta)^{x_k} (1 - p_{k|r}(\theta))^{1-x_k} \right]^{z_{kr}} \ , \tag{6}$$

where q_{kr} is the probability that cell k is assigned type $r \in \mathcal{R}$ of potential types and z_{kr} is an indicator variable for the assignment of k to r. The complete data log likelihood is then

$$\ell(X, Z; \theta) = \sum_{k \in \bar{\mathcal{W}}(t), r \in \mathcal{R}} z_{kr} \left[\log q_{kr}(\theta) + x_k \log p_{k|r}(\theta) + (1 - x_k) \log(1 - p_{k|r}(\theta)) \right] . \tag{7}$$

Of course, in practice we don't observe Z, and cannot maximize the complete data log likelihood directly. Instead, we consider a version of the EM algorithm in which we first formulate a belief over Z and then take the expectation of the expected log likelihood with respect to that belief. The resulting updates may be formulated as

1. **E-Step.** Compute the expected value of z_{kr} for each k and r with respect to the current parameters $\hat{\theta}$, which are simply

$$\gamma_{kr} = q_{rk}(\hat{\theta}). \tag{8}$$

2. **M-Step.** Maximize the expected complete data log-likelihood, given by

$$U(\theta|\hat{\theta}) = \sum_{k \in \bar{\mathcal{W}}(t), r \in \mathcal{R}} \gamma_{kr} \left[\log q_{kr}(\theta) + x_k \log p_{k|r}(\theta) + (1 - x_k) \log(1 - p_{k|r}(\theta)) \right] . \tag{9}$$

Executing the M-step requires computation of the gradient $\nabla_\theta U(\theta|\hat{\theta})$. We have

$$\nabla_\theta U(\theta|\hat{\theta}) = \sum_{k\in\bar{\mathcal{W}}(t), r\in\mathcal{R}} \gamma_{kr} \left[\frac{\nabla_\theta q_{kr}(\theta)}{q_{kr}(\theta)} + \left(\frac{x_k}{p_{k|r(\theta)}} - \frac{1-x_k}{1-p_{k|r}(\theta)} \right) \nabla_\theta p_{k|r}(\theta) \right] . \tag{10}$$

Completing the M step thus requires the gradients $\nabla_\theta q_{kr}(\theta)$ and $\nabla_\theta p_{k|r}(\theta)$. The former can be computed in terms of the latter as

$$\frac{\nabla_\theta q_{kr}(\theta)}{q_{kr}(\theta)} = \nabla_\theta \log p_{k|r}(\theta) - \nabla_\theta \log \sum_{r\in\mathcal{R}} p_{k|r}(\theta) \tag{11}$$

$$= \frac{\nabla_\theta p_{k|r}(\theta)}{p_{k|r}(\theta)} - \frac{\sum_{r\in\mathcal{R}} \nabla_\theta p_{k|r}(\theta)}{\sum_{r\in\mathcal{R}} p_{k|r}(\theta)} , \tag{12}$$

which is tractable when $|\mathcal{R}|$ is small. The functional forms of the gradients depend on which of (4) and (5) are used. In both models, we have

$$\frac{\partial p_{k|r}(\theta)}{\partial \alpha_s} = \begin{cases} \frac{p_{k|r}(\theta)-\beta_r}{\alpha_r} & s = r \\ 0 & \text{otherwise} . \end{cases} \tag{13}$$

and

$$\frac{\partial p_{k|r}(\theta)}{\partial \beta_s} = \begin{cases} 1 & s = r \\ 0 & \text{otherwise} . \end{cases} \tag{14}$$

In the case of (4), the derivative with respect to γ takes a somewhat unpleasant form; using the chain rule, we obtain

$$\frac{\partial p_{k|r}(\theta)}{\partial \gamma_r} = \gamma_r \frac{\left[\left(\sum_{j\in G} d_{jk}^{-(\gamma_r+1)} \right) \left(\sum_{j\in\mathcal{W}_r} d_{jk}^{-\gamma_r} \right) - \left(\sum_{j\in G} d_{jk}^{-(\gamma_r)} \right) \left(\sum_{j\in\mathcal{W}_r} d_{jk}^{-(\gamma_r+1)} \right) \right]}{\left(\sum_{j\in G} d_{jk}^{-\gamma_r} \right)^2} \tag{15}$$

$$= \gamma_r \frac{\left[\left(\sum_{j\in G} d_{jk}^{-(\gamma_r+1)} \right) \left(p_{k|r}(\theta) - \beta_r \right) \alpha_r^{-1} - \left(\sum_{j\in\mathcal{W}_r} d_{jk}^{-(\gamma_r+1)} \right) \right]}{\sum_{j\in G} d_{jk}^{-\gamma_r}} \tag{16}$$

Using (5), we get something a bit more tractable:

$$\frac{\partial p_{k|r}(\theta)}{\partial \gamma_r} = \frac{-\gamma_r}{\alpha_r} \left(p_{k|r}(\theta) - \beta_r \right) \left(\alpha_r - \left(p_{k|r}(\theta) - \beta_r \right) \right) \sum_{j\in\mathcal{W}_r} d_{jk}^{-(\gamma_r+1)}, \tag{17}$$

which involves fewer and smaller summations.

2 One Unsupervised Model

Our third model under consideration views the inference problem as an *unsupervised* problem in which we seek to model the density of new settlements via a mixture of densities centered at existing settlements. For each newly-settled site k, there is an unobserved parent cluster j composed of spatially-adjacent sites in $\mathcal{W}(t)$. We view the new settlements as generated by a probability distribution

$$P(j;\theta) = \sum_k \pi_j p_j(k;\theta_j), \tag{18}$$

where $P(k;\theta)$ is the pdf at site k, $p_j(k;\theta_j)$ is the contribution to P of existing settlement j at site k, and π_j are mixing coefficients satisfying $\sum_j \pi_j = 1$ and $\pi_j \geq 0$. The distributions $p_j(k;\theta_j)$ reflect the shape of the contribution of each existing cluster, while the mixing coefficients π_j determine their relative importance. For example, if a city is nearly monocentric and compact in space, we might expect that the distribution $p_j(k;\theta_j)$ corresponding to the center to be highly concentrated in space, and that the associated mixing coefficient π_k would be high. In contrast, if a city is polycentric, we might observe multiple concentrated distributions with lower mixing coefficients.

To perform inference in this model, we develop an EM algorithm that is highly reminiscent of standard Gaussian mixture-modeling. The hidden data is the matrix Z of parent-child settlement relationships, with $z_{kj} = 1$ if settled site k is "descended" from cluster j. The complete data likelihood is then

$$L(X,Z;\theta) = \prod_{k,j} (\pi_j p_j(k;\theta_j))^{z_{kj}} \ , \tag{19}$$

giving a log-likelihood

$$\ell(X,Z;\theta) = \sum_{k,j} z_{kj} \left[\log \pi_j + \log p_j(k;\theta_j)\right] \ . \tag{20}$$

Since we don't observe the complete data, we instead estimate the expected values of z_{kj} as

$$\gamma_{kj} = \frac{\pi_j p_j(k;\theta_j)}{\sum_j \pi_j p_j(k;\theta_j)} \tag{21}$$

and then maximize the expected log-likelihood

$$\ell(X,Z;\theta) = \sum_{k,j} \gamma_{kj} \left[\log \pi_j + \log p_j(k;\theta_j)\right] \ . \tag{22}$$

with respect to the mixture coefficients π_j and the component parameters θ_j. Since the mixture coefficients are uncoupled from the component parameters, we can obtain expressions for them in closed form, giving

$$\pi_j = \frac{1}{N} \sum_k \gamma_{kj} \tag{23}$$

which expresses the mixture coefficients as total responsibilities. The component parameters θ_j need to be optimized iteratively depending on the functional form of p_j. One appropriate possibility that captures the flavor and potentially the physics of Rybski-style growth is

$$p_j(k;\gamma_j) = \frac{1}{Z(\gamma_j)} \sum_{i\in C_j} d_{ik}^{-\gamma_j}\ , \tag{24}$$

where $Z(\gamma_j)$ is the normalizing partition function. To optimize (22), we need the gradient of $\log p_j(k;\theta_j)$, which we obtain as

Would it make more sense to exponentiate and take advantage of the nice properties of exponential families? Not linear, so unclear whether this would be helpful.

$$\begin{aligned}
\nabla_{\theta_j} \log p_j(k;\theta_j) &= \frac{1}{p_j(k;\theta_j)} \nabla_{\theta_j} p_j(k;\theta_j) \qquad (25)\\
&= \frac{1}{p_j(k;\theta_j)} \frac{\left(-\gamma_j \sum_{i\in C_j} d_{ik}^{-(\gamma_j+1)}\right) Z(\gamma_j) - \nabla Z(\gamma_j) \sum_{i\in C_j} d_{ik}^{-\gamma_j}}{Z(\gamma_j)^2} \qquad (26)\\
&= \frac{1}{p_j(k;\theta_j)} \frac{\left(-\gamma_j \sum_{i\in C_j} d_{ik}^{-(\gamma_j+1)}\right) - p_j(k;\gamma_j) \nabla Z(\gamma_j)}{Z(\gamma_j)}, \qquad (27)
\end{aligned}$$

which is not especially pleasant to compute but also likely doesn't need to be computed too many times.

3 Todo

1. Model Analyses
   - Model capacity
   - Fisher information (parameter identifiability)
2. Plate viz of models