EISI Notes – Tom Dietterich – July 14, 2014

Preference Modeling

I met with Kaitlin and Ivan and we discussed a variant of the multinomial model that Rebecca and I have been exploring with our honors student Sneha Krishna. The basic idea is to posit that for each pollinator species b each flower species f has a fixed "desirability" $d_b(f)$. For each combination of meadow m and day t, let $A_{mt}(f)$ be the total number of ("active") flower-bearing stems for species f summed across all plots in the meadow. Let $V_{mt}(b,f)$ be the number of visits by species b to flower f in meadow m on day t summed across all plots. Let $V_{mt}(b) = \sum_f V_{mt}(b,f)$ be the total number of visits of species b to *any* flower species summed across all plots. Then this model treats these $V_{mt}(b)$ visits as iid draws from a multinomial distribution defined as

$$P(b \to f|t,m) = \frac{A_{mt}(f)d_b(f)}{\sum_{f'} A_{mt}(f')d_b(f')}.$$

In words, we expect pollinators of species b to visit flowers of species f in proportion to the product of f's desirability and f's abundance. Note that if a given flower species f is absent (or not flowering) in meadow m on day t, then it does not "participate" in this multinomial distribution and the probability of visiting it is zero.

Let $D = \left(d_b(f_1), \ldots, d_b(f_F)\right)$ be the collection of desirabilities for species b and all flower species $1, \ldots, F$.

The likelihood function is

$$\mathcal{L}(D) = \prod_m \prod_t \binom{V_{mt}(b)}{V_{mt}(b,f_1) \quad \ldots \quad V_{mt}(b,f_F)} P(b \to f_1|t,m)^{V_{mt}(b,f_1)} \times \cdots \times P(b \to f_F|t,m)^{V_{mt}(b,f_F)}.$$

I proposed to maximize this via gradient ascent in the d_b parameters.

A further extension would be to model the desirability as a function of covariates based on traits of the pollinators and the flowers. For example we could have $d_b(f) \propto \exp \beta_0 + \beta_1 x_1(b,f) + \cdots \beta_J x_J(b,f)$, where the covariates $x_j(b,f)$ capture some aspect of the compatibility between b and f or the utility of nectar from f for b.

When Jorge and I were talking with Ivan and Kaitlin, it was suggested that this problem could be divided into two parts: (a) estimating D and (b) predicting D from the covariates. This would be a good initial strategy, but if we really believe the covariate model, then the β's should be fit to maximize the likelihood ("end to end"). This will allow the covariates to "pull" the desirability values toward values that are easier to explain using the covariates.

Analyzing the Structure of the Plant-Pollinator Visit Matrix

After lunch, I met with Jacob, Ryan, Lauren, and Disa. The conversation centered on different approaches to analyzing/decomposing/factoring the plant-pollinator visit matrix and possibly filling in missing links (i.e., hypothesizing unobserved links). For this purpose, I think it would be good to use the number of visits $V_{mt}(b, f)$ rather than just the visit/not visit binary version.

Jorge introduced the very useful distinction between the *possible* matrix $\mathbb{P}$, the *actual* matrix $\mathbb{A}$, and the *observed* matrix $\mathbb{O}$. The goal of this discussion was to try to understand $\mathbb{A}$ by analyzing $\mathbb{O}$.

Idea 1: Explore matrix completion methods (e.g., from collaborative filtering).

The collaborative filtering methods developed for the Netflix challenge apply the Singular Value Decomposition to factor the matrix (in various ways). There is also a large literature on non-negative matrix factorization. I don't know much about this literature, but the basic idea is to (approximately) decompose the observed matrix into a product of two or more smaller matrices in order to minimize the squared reconstruction error (i.e., the squared error between the reconstructed and observed entries in the matrix). The reconstructed matrix can be viewed as a "filled in" version of the original matrix. The main thing I don't understand is that we only want to minimize the error on the non-zero cells in the observed matrix, since we don't trust our zeroes. However, we presumably want to encourage the algorithm to reconstruct most of the observed zeroes as real zeroes. So I'm thinking that maybe the objective function has two components: one for the observed non-zeroes and one for the observed zeroes. We put higher weight on the first component and lower weight on the second. We could explore how varying that lower weight influences the resulting matrix. I believe Jacob is interested in pursuing this general idea. In our discussions, the issue of the zeroes was not addressed in much depth – my notes above are afterthoughts.

Idea 2: Explore double-clustering methods (from document clustering and DNA microarray analysis)

The idea here is to explore whether the pollinator species can be clustered into a set of clusters and the flower species can be clustered into a set of clusters so that if there is a link from bee cluster i to flower cluster j then we predict that *all* bees in cluster i will visit *all* flowers in cluster j. We did some web searching and found the R package **biclust** which implements several different biclustering algorithms. So the main plan here is for someone to explore those different algorithms. The one that caught my eye was the BCSpectral method, which uses a form of spectral clustering. An algorithm that is not implemented in biclust is the agglomerative information bottleneck method, so if none of the biclust methods gives adequate results, someone could implement that instead.

It is possible to view biclustering as matrix factorization. We factor the observed matrix into three matrices: (a) one that maps a bee species to a bee cluster, (b) the cluster-to-cluster adjacency matrix, and (c) a matrix that maps the flower cluster to the individual flower species.

My understanding was the Ryan and Lauren were interested in exploring this approach.

Idea 3: Learn to predict interactions from traits

A third topic that was discussed at various points with various people would be to fit a predictive machine learning model to predict the 0's and 1's in the (binary) 𝕆 matrix based on traits of the flower and traits of the pollinator. Vera provided the trait spreadsheet ConeCoVarioatesRebeccaConePeak2011.xlsx, which I put into the dropbox folder under a "Traits" folder.

I see two possible approaches here. One would be to spend some time on "feature engineering" to develop a set of features that summarize the "compatibility" between the flower and the pollinator. Then we could apply logistic regression or something similar. The second approach would be to just concatenate the existing traits and apply gbm (boosted trees) to try to make the predictions. The advantage of gbm is that it could automatically discover the relevant feature combinations. That would be easier, but probably would not work as well.

We should also think carefully about the "zeroes problem". It occurs to me that perhaps we should view the zeroes as "pseudo-absences" and apply maxent (or the equivalent Inhomogeneous Poisson Process model; or Trevor Hastie's extreme-weighted logistic regression) instead of standard supervised learning to fit the data and make predictions. We discovered from web searches that there is an R package, **dismo**, that wraps Steven Phillip's maxent code.

I don't think anyone jumped at this problem, but maybe Lauren (or Jacob) might be interested.