Imputing time series data with latent factors

Adam Coogan

February 13, 2017

1 Algorithm

Let the data $t \times N$ matrix for a cluster with N sensors at a given time t be

$$D^{1:t} = \begin{pmatrix} d_{1\ 1} & \dots & d_{1\ N} \\ \vdots & \ddots & \vdots \\ d_{t\ 1} & \dots & d_{t\ N} \end{pmatrix}.$$

Some of the data points will be missing. My approach is to fill in missing data in the last row of $D^{1:t}$ by applying a static "recommender system" algorithm to the last Δt rows of the matrix, $D^{t-\Delta t:t}$. This simple prescription turns any static recommender into one that can be applied online. The algorithm runs online (ie, each time a new observation containing missing data comes in).

I use a latent factor model to fill in the missing data. The idea is that while the dimension of $D^{t-\Delta t:t}$ is $\Delta t \times N$, its rank is much lower since the sensors are all sampling the same weather patterns. We therefore want to find the low rank approximation

$$D^{t-\Delta t:t} \approx UV^T,$$

where U is a $\Delta t \times k$ matrix, V is a $N \times k$ matrix and $k \ll \min(\Delta t, N)$. k is the number of latent factors in the model. Singular value decomposition is of course one method for finding U and V. While SVD is optimal when all of the data is observed, it is not necessarily optimal when there is missing data.

U and V can be found by solving minimizing an objective function:

$$U, V = \arg\min_{\tilde{U},\tilde{V}} f(\tilde{U}, \tilde{V}),$$

$$f(\tilde{U}, \tilde{V}) \equiv \left[\sum_{(i,j)\in\mathcal{O}} \left(D_{ij}^{t-\Delta t:t} - (\tilde{U}\tilde{V}^T)_{ij} \right)^2 \right],$$

where $\mathcal{O}$ is the set of indices (i, j) for which $D_{ij}^{t-\Delta t:t}$ is observed. This is simple to solve using gradient descent. As initial guesses, I choose each element of U and V from

$$\mathcal{N}\left(\frac{\mu_{t-\Delta t:t}}{\sqrt{|\mu_{t-\Delta t:t}|}}, \frac{\sigma_{t-\Delta t:t}}{\sqrt{|\mu_{t-\Delta t:t}|}} \right),$$

where $\mu_{t-\Delta t:t}$ and $\sigma_{t-\Delta t:t}$ are the mean and standard deviation of the observed entries in $D^{t-\Delta t:t}$. The normalization ensures that U and V's elements change by a similar amount during each step of gradient descent. Since the problem is nonconvex, the objective function has many local minima; the imputation algorithm should therefore be run multiple times and the resulting imputed values averaged.

Since the optimization problem is underdetermined, it is important that k be small. k can be set by cross-validation, where the validation set is created by removing data for a single sensor from times during which few observations are missing. I've found that $k = 1$ works well. The other parameter in the model is the window size, Δt, and can also be set using cross-validation.

The algorithm has issues if a whole column of $D^{t-\Delta t:t}$ is missing (ie, if sensor i has no observations over the window). In this case it is impossible to determine the corresponding row $V_{i1}, \ldots, V_{ik}$ in V which specifies how U's columns scale and sum to form the sensor's time series. The best solution I've come up with in this situation is to add in the last imputed (or observed) data point for the sensor, which is $D^{1:t}_{t-\Delta t-1:i}$. This biases the algorithm towards past observations and can lead to numerical instability.

Algorithm 1 Latent factor imputation

$\hat{D}^{t-\Delta t:t} \leftarrow$ last Δt rows of $D^{1:t}$

for $s = 1, \ldots, N$ **do**
    if $\hat{D}^{t-\Delta t:t}_{1s}, \ldots, \hat{D}^{t-\Delta t:t}_{\Delta t s} = \text{nan}$ **then**
        $\hat{D}^{t-\Delta t:t}_{1\ s} \leftarrow D^{1:t}_{t-\Delta t-1:s}$
    end if
end for

$\mathcal{O} \leftarrow$ set of indices for which data was observed
$\mathcal{M} \leftarrow$ set of indices for which data is missing
$n_{\text{LF}} \leftarrow$ number of latent factors

$n_{\text{runs}} \leftarrow$ number of times to run gradient descent
$\hat{A} \leftarrow \Delta t \times N$ matrix of zeros to hold average of gradient descent solutions

for $i = 1, \ldots, n_{\text{runs}}$ **do**
    $U, V \leftarrow$ generate initial guesses as described above
    $U, V \leftarrow$ get gradient descent solution of $\arg\min_{U,V} f(U, V)$
    $\hat{A} \leftarrow \hat{A} + \frac{1}{n_{\text{runs}}} UV^T$
end for

for $(i, j) \in \mathcal{M}$ **do**
    $\hat{D}^{t-\Delta t:t}_{ij} = \hat{A}_{ij}$
end for

return $\hat{D}^{t-\Delta t:t}$