Online supplementary information for the paper

A method for learning a sparse classifier in the presence of missing data for high-dimensional biological datasets

Kristen Severson\(^1\), Brinda Monian\(^1\), J. Christopher Love\(^1\), and Richard D. Braatz\(^1\)

\(^1\)Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

1 Derivation of the EM steps

The EM-SDA algorithm uses a classic application of the EM algorithm [1]. For the case where there is no missing data, the complete data log-likelihood is

\[
\ell(W, \mu, \Delta, \sigma^2|x_i, t_i, \tau, \gamma, y_i) = \ln p(t_i) + \ln p(x_i|t_i, W, \mu, \Delta, \sigma^2) + \ln p(\Delta|T) + \sum_{j=1}^{p} \ln p(\tau_j|\gamma) \tag{1}
\]

The E-step requires the conditional distribution of the unobserved variables given the observed variables and the current values of the parameters. Given the conditioning set, the distribution of \(\tau\) is independent of the distribution of \(t\):

\[
q_i(t_i, \tau) = p(t_i|x_i, W, \mu, \Delta, \sigma^2) \prod_{j=1}^{p} p(\tau_j|\gamma, \Delta) \tag{2}
\]

where \(\tau_j^2\) only appears in the complete data log-likelihood as its inverse in terms that involve the parameters. Therefore, we are only concerned with \(\langle 1/\tau_j^2 \rangle\), conditioned on the current values of \(\Delta_j\) and \(\gamma\).

[7] derive the expression

\[
p(1/\tau_j^2|\Delta, \gamma) = IG\left(\sqrt{\frac{\gamma^2}{\Delta_j^2}}, \gamma^2\right) \tag{3}
\]

where IG is the inverse gamma distribution.

The expected complete data log-likelihood function is formed using the posterior distribution \(q_i(t_i, \tau)\) and the complete log-likelihood function

\[
E[\ell(W, \mu, \Delta, \sigma^2|x_i, t_i, \tau, \gamma, y_i)] = \sum_{i=1}^{n} \int \int q_i(t_i, \tau) \left(\ln p(t_i) + \ln p(x_i|t_i, W, \mu, \Delta, \sigma^2) + \ln p(\Delta|T) \right.
\]

\[
+ \sum_{j=1}^{p} \ln p(\tau_j|\gamma) \bigg) \right) dt_i dx_i^m d\tau \tag{4}
\]

Using the factorization of \(q_i\), the definitions of the distributions, and the dropping of terms that do not depend on \(\theta\), this equation can be rewritten as

\[
E[\ell(W, \mu, \Delta, \sigma^2|x_i, t_i, \tau)] \propto \sum_{i=1}^{n} -\frac{1}{2} \ln(|\sigma^2 I_p|) - \frac{1}{2} E_{q_i(t)}[\Delta^\top T^{-1} \Delta]
\]

\[
- \frac{1}{2\sigma^2} E_{q_i(t)}[(x_i - \mu - \Delta - Wt_i)^\top (x_i - \mu - \Delta - Wt_i)] \tag{5}
\]
The resulting E- and M-steps are provided in the text in eqns. 4abc, 5abcd, and 6.

The natural update equation for \( \Delta^{\text{new}} \) is

\[
\Delta^{\text{new}} = (I_p + \sigma^2T^{-1})^{-1} \frac{1}{n} \sum_{i=1}^{n} x_i - \mu - W(t_i)
\]  

(6)

However, when implementing the update step, \( T^{-1} = \text{diag}((1/\tau_j)) = \text{diag}(\gamma/|\Delta_j|) \) would have a numerical issue since many elements of \( \Delta \) are expected to go to zero. To avoid this numerical issue [3, 6], the alternative update equation

\[
\Delta^{\text{new}} = T(\sigma^2I_p + T)^{-1} \frac{1}{n} \sum_{i=1}^{n} x_i - \mu - W(t_i)
\]  

(7)

is implemented [9].

2 Distribution of \( t \) and \( x^m \)

To describe the joint distribution of \( t\) and \( x^m\), a new variable

\[
z_i = \begin{bmatrix} t_i \\ x^m_i \end{bmatrix},
\]

(8)

is defined where \( p(z_i|x^o_i, W, \sigma^2, \mu, y_i) \) is a Gaussian distribution described by the information form of the multivariate Gaussian distribution,

\[
\Lambda_z = \begin{bmatrix}
I_a + \frac{1}{\sigma^2} W^T W & -\frac{1}{\sigma^2} W^m W^T \\
-\frac{1}{\sigma^2} W^m W^T & \frac{1}{\sigma^2} I_m
\end{bmatrix}
\]

(9)

\[
\eta_z = \begin{bmatrix}
\frac{1}{\sigma^2} W^o (x^o - \mu^o - \Delta^o) - \frac{1}{\sigma^2} W^m (\mu^m + \Delta^m)
\end{bmatrix}
\]

(10)

Using these factors, the mean and covariance of the posterior distribution can be defined by

\[
\Sigma_z = \Lambda_z^{-1} = \begin{bmatrix}
\sigma^2 (\sigma^2 I_a + W^o W^o)^{-1} \\
\sigma^2 (\sigma^2 I_a + W^o W^o)^{-1} W^m W^T \\
\sigma^2 (I_m + W^m (\sigma^2 I_a + W^o W^o)^{-1} W^m W^T)
\end{bmatrix}
\]

(11)

\[
\mu_z = \Sigma_z \eta_z = \begin{bmatrix}
(\sigma^2 I_a + W^o W^o)^{-1} W^o (x^o - \mu^o)
\end{bmatrix}
\]

(12)

3 Convergence and cross-validation

To test for convergence of the algorithm, the observed data negative log-likelihood (NLL) should be monitored. The observed data NLL is

\[
\ell = \sum_{i=1}^{n} \left[ \frac{|o|}{2} \ln 2\pi + \frac{1}{2} \ln |W^o W^o^T + \sigma^2 I_{|o|}| \\
+ \frac{1}{2} (x^o - \bar{\mu}^o - \Delta^o)^T (W^o W^o^T + \sigma^2 I_{|o|})^{-1} (x^o - \bar{\mu}^o - \Delta^o) \\
+ \frac{p}{2} \ln 2\pi - p \ln \frac{\gamma^2}{2} + \frac{1}{2} \sum_{k \in K} \sum_{j=1}^{p} \ln \tau_j + \frac{\Delta_j^2}{\tau_j} + \gamma^2 \tau_j^k \right]
\]

(13)

As the algorithm proceeds, many of the elements of \( T = \text{diag}(\tau_j) \) will go to zero, which represents a change in the degrees of freedom that needs to be reflected in the observed data NLL. The value of \( p \) should correspond to the
length of the non-zero elements along the diagonal of $T$. Additionally, only the corresponding values of $\Delta$ should be used.

To choose the values of the latent dimension $a$ and the sparsity tuning parameter $\gamma$, a cross-validation strategy is recommended. The training dataset is partitioned into two parts: a $1/k$ proportion of the dataset for validation and the remaining data for training. For the presented work, $k$ was selected as 5. The performance on the held-out validation set is used to select the values. Fig. S1 shows an example as applied to the Golub et al. dataset.

### 4 Simulation

Details on the simulation study (Section 3.1) are provided here. Each dataset has 100 ‘experiments’ and 2000 ‘measurements’. Half of the experiments were assigned to class 1 and the other half were assigned to class 0. The data are modeled using class-specific means for the discriminating variables and zero means for the remaining data. The data are modeled using a shared covariance with a latent dimension of 5. The error and factor loadings, as described in eqn. 1, are scaled to control class overlap. To determine the covariance matrix, first a matrix of size $p \times a$ is generated where each element is uniformly distributed between $[-1, 1]$. The QR decomposition is applied to orthogonalize the result which is then rescaled. The full covariance matrix is calculated by $\Sigma = WW^\top + \sigma^2 I_p$. The rescaling and value of $\sigma^2$ are selected to meet the overlap criteria. This step specifies the class-specific means and covariance. The observed data are then simulated using these parameters as a multivariate normal distribution, conditioned on the class.

### 5 Biomedical datasets

The detailed information on dataset size and preprocessing is presented here. In [4], the goal is to classify two types of leukemia – acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) – using gene expression data. The dataset contains 38 training samples (27 ALL) and 34 testing samples (20 ALL). Each sample has 7129 gene measurements and the preprocessing methodology described by [2] was followed. For [8], one of the several presented classification problems was chosen, specifically classifying patients with E. coli infections from patients with S. aureus infections using gene expression data. The dataset contains 20 training samples (10 E. coli) and 39 testing samples (18 E. coli). Each sample has 211 gene measurements and the preprocessing methodology described by [8] was followed. For [5], one of the several presented classification problems was chosen, specifically classifying rescued and failed learning in trisomic mice using protein expression levels from reverse phase protein
arrays. The dataset contains 120 training samples (67 rescued learning) and 120 testing samples (68 rescued learning). Each sample has 77 protein measurements and the preprocessing methodology described by [5] was followed. In all cases, pre-processing is applied after missingness is introduced.

6 Biological analysis

Biological significance of each classifier was assessed by a score derived from Pubmed search results. The score, as presented in Fig. 4, S2, and S3, is the number of results for “gene/protein name” + “problem domain” squared divided by the total number of results for the gene/protein. The problem domain terms are: leukemia, infection, and Down syndrome/memantine/cognitive where “/” refers to OR statements. The score is then log-scaled. A “–” indicates that there were no results for that gene/protein.

References


Figure S2: Genes that were selected for the various infection cases. NSC and SDA are combined with BPCA for imputation. Shaded cells indicate that a particular gene was selected and the intensity of the cell represents the infection-relevant score based on an independent literature review. The dashes represent genes that were selected but were not found during the literature review.
Figure S3: Proteins that were selected for the various trisomic mice cases. NSC and SDA are combined with KNN for imputation. Shaded cells indicate that a particular protein was selected and the intensity of the cell represents the trisomy-relevant score based on an independent literature review.