Tools in Multivariate Analysis

Kenneth Tay

Jul 18, 2018

- Given n observations, find some $k \ll n$ prototypes/objects to represent them (or their variation). Sometimes this amounts to fitting a low-dimensional surface to the observations.
  - General algorithms: principal components analysis (PCA), factor analysis, projection pursuit, independent component analysis (ICA), principal curves and surfaces
  - Rows in contingency tables: correspondence analysis
- Matrix completion: Hard Impute, Soft Impute
- Classification: Discriminant analysis (all versions)
- Given distances/dissimilarities/similarities, find some lower-dimensional embedding that preserves this structure:
  - General algorithms: classical metric scaling, Kruskal-Shepard metric scaling, Kruskal-Shepard non-metric scaling.
  - Focus on local structure: isometric feature mapping (ISOMAP), local linear embedding (LLE), local MDS
- Unsupervised clustering: k-nearest neighbors, k-means, self-organizing maps (SOM), spectral clustering

Method	**Description & Assumptions**	**Pros & Cons**
Gaussian copulas	Idea: Want to draw samples from some multivariate distribution F that has marginals $F_1, \ldots, F_p$. We can use a multivariate Gaussian to do so in a way that respects the marginal distributions and the correlations between the features. We assume $Z \sim N_p(0, R)$, where R is some correlation matrix. Set $X_j = F_j^{-1}(\Phi(Z_j))$. Then $(X_1, \ldots, X_p)$ will have the desired marginals with some correlation between the features.	- Have to estimate R. Also, R gives correlation between the Z_j's, not the X_j's.
Principal components analysis (PCA)	Dimensionality reduction method. Idea: Think of observations as points in $\mathbb{R}^p$. For a given k, find the top k orthogonal directions along which the observations vary the most. This can be accomplished simply by taking an SVD of the data matrix: if $X = UDV^T$, then the first k PCs are given by $U_k D_k$, and the first k loading vectors are given by V_k.	+ Easy to compute + Makes intuitive sense as a dimensionality reduction tool. - PCs are in general linear combinations of all p original features, so not sparse in original feature space. (This can be fixed by using sparse PCA methods.) - How to choose the number of PCs?
Hard Impute	For matrix completion with missing entries. Let Ω denote the set of entries of X that are observed. Idea: Assume some low rank structure, minimize Frobenius norm over observed entries: $\min_{\text{rank}(Z)=L} \|P_\Omega(X) - P_{\Omega^\perp}(Z)\|_F$. Iterative algorithm: Initialize by randomly filling in the missing entries. In each iteration, take the rank-L SVD of the most updated X matrix, then update the missing entries in X with the entries from this rank-L SVD.	+ Fast algorithm. - Assumes low rank structure. - Objective function is non-convex, so algorithm is not guaranteed to converge to a global minimum.
Soft Impute	Idea: Solve a convex relaxation of the minimization problem for Hard Impute instead: $\min_{\text{rank}(Z)=L} \|P_\Omega(X) - P_{\Omega^\perp}(Z)\|_F^2 + \lambda\|Z\|_*$, where $\|\cdot\|_*$ denotes the nuclear norm. Algorithm is basically the same as Hard Impute, except instead of taking the rank-L SVD $Z^{i+1} = U_L D_L V_L^T$, take $Z^{i+1} = U_L \mathcal{S}(D, \lambda)_L V_L^T$, where $\mathcal{S}(d, \lambda) = (d - \lambda)_+$.	+ Problem is convex and so we can prove convergence. - Not the objective function that we really want.

Method	**Description & Assumptions**	**Pros & Cons**
Graphical LASSO	Assume that the variables $X_1, \ldots, X_p$ are jointly Gaussian with joint density $X \sim \mathcal{N}(\mu, \Sigma)$. Let $\Theta = \Sigma^{-1}$. In this set-up, X_i and X_j are conditionally independent iff $\Theta_{ij} = 0$. Idea: Estimate conditional dependence structure of data by using L_1 regularization of the log-likelihood: $\max_{\Theta} \log\det\Theta - \mathrm{tr}(S\Theta) - \lambda\|\Theta\|_1$, where S is the sample covariance.	
Factor analysis	Idea: Produce a small set of factors which explain the correlations among the given variables. The model is $X = \Lambda f + e$, where X represents the observed variables, $e \in \mathbb{R}^p$ represents the unique factors for each variable, $f \in \mathbb{R}^q$ represents the common factors, and $\Lambda \in \mathbb{R}^{p\times q}$ represents the factor loadings. By considering the covariances, we get $\Sigma = \mathrm{Cov}(X) = \Lambda\Lambda^T + \Psi$, where $\Psi = \mathrm{Cov}(e) = \mathrm{diag}(\psi_1, \ldots, \psi_p)$. Various methods are used to estimate Λ and Ψ.	+ There are factor analysis methods that do not have any distributional assumptions (e.g. principal factor method); they just work on correlations. - For any decomposition Λ and Ψ, $V\Lambda$ and Ψ (with $V \in \mathbb{R}^{q\times q}$ orthonormal) give an equivalent model. Hence, there is an inherent non-uniqueness for factor analysis.
Projection pursuit	Idea: For multivariate random vector y, most projections $\alpha^T y$ (with $\|\alpha\|_2 = 1$) look "normal". We try to find projections which are "non-normal". These projections can show us some of the structure of teh data. Defining entropy as $I(f) = -\mathbb{E}_f[\log f]$, the more random or uniform a distribution, the higher the entropy. Thus, we want to find α such that $I(\alpha^T y)$ is minimized. Friedman formulates the problem as maximizing a quantity representing departure from uniform instead: $\min_{\|\alpha\|_2=1} \int_{-1}^{1}[P_R(r) - 1/2]^2 dr$, where P_R is the density of $R = 2\Phi(\alpha^T y) - 1$.	
Independent component analysis (ICA)	Idea: Our data X is really a linear transformation of sources S, $X = AS$, with the elements of S being independent and non-Gaussian. A is known as the mixing matrix. Our goal is to estimate A and the distributions of the S_j's. Usually solved using entropy H and mutual information $I(Y) = \sum_{j=1}^{p} H(Y_j) - H(Y)$. We want to find A that minimizes $I(A^T X)$. There is also an alternating algorithm (ProDenICA) using tilted Gaussian densities.	+ Unlike factor analysis, there is a unique solution.

Method	**Description & Assumptions**	**Pros & Cons**
Correspondence analysis	Idea: Try to perform PCA for $J \times K$ contingency tables. After normalizing by row totals, each row is a "profile" in the simplex in $\mathbb{R}^K$ (entries sum to 1). We want to find a subspace that approximates the rows well in the appropriate metric. The solution to this problem ends up being the generalized SVD.	
Principal curves & surfaces	Goal is to find a low-dimensional manifold which approximates the data well. Idea: PCA solves min $\sum_{i=1}^n \|x_i-(\alpha_0+V\gamma_i)\|^2$. Instead of approximating with a linear manifold, approximate by a smooth manifold: $\min_{f,\gamma_i} \sum_{i=1}^n \|x_i-f(\gamma_i)\|^2$, where f belongs to some smooth family. Solve using an iterative algorithm: For fixed f, for each i pick γ_i to minimize $\|x_i - f(\gamma_i)\|$. For fixed γ_i's, model $x_{ij} = f_j(\gamma_i) + \epsilon_{ij}$.	+ Typically used for data visualization (2D & 3D).
K-means clustering	Idea: minimize the within-cluster scatter: $\sum_{k=1}^K \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$, where $C(i)$ is the cluster membership for observation i. Can be solved iteratively: Given centroids, assign each observation to its closest centroid. Given assignments, recompute centroid locations.	+ Easy to implement. - Solution depends on starting configuration (only local optimum reached). - How to choose K?
Self-organizing maps (SOM)	An online version of K-means, where the centroids are somewhat constrained. As points come in, add point to the cluster whose centroid is closest to it. Then move the cluster centroid closer to the point (based on a learning rate parameter α), and move other centroids which are connected to this centroid closer as well.	+ Online algorithm, so can be updated as new points come in. - Have to deal with two metrics: one to measure distances between observations, one to measure distances between centroids. - Have to choose number of centroids.

Method	**Description & Assumptions**	**Pros & Cons**
Linear discriminant analysis (LDA)	Supervised learning: To determine a classification rule for observations in $\mathbb{R}^p$ into k groups. Idea: Assume that for each group j, $X \mid$ in group $j \sim \mathcal{N}(\mu_j, \Sigma)$, with the covariance Σ being the same across groups. Assume marginal probabilities $P(\text{group } j) = \pi_j$. Parameters π_j, μ_j and Σ are estimated by maximum likelihood. For new data x^*, compute the discriminant functions $\log P(\text{in group } j \mid x^*) = \log \pi_j + (x^*)^T \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j^T \Sigma^{-1} \mu_j$, and classify to the group with the largest value. Results in linear boundaries between the classes.	+ Computation is very easy. - Model has linear boundaries: may be too simple. - Performance depends on the validity of the Gaussian distribution assumption.
Reduced rank discriminant analysis	Idea: In LDA, the k centroids lie on an $(k-1)$-dimensional hyperplane. Projecting points onto this hyperplane does not change the classification rule. Let $A \in \mathbb{R}^{p \times (k-1)}$ be the first $k-1$ eigenvectors of $W^{-1}B$ (defined in ESL p114, its columns span the space containing the k centroids). After sphering the data, we can project our points onto this space ($x \mapsto A^T x$), and assign it to the nearest centroid (adjust for prior probabilities). We can do even further dimensionality reduction: to constrain the centroids to lie on an r-dimensional hyperplane, just take the first r columns of A.	+ Can be used as a data reduction tool. When $r = 2$ or 3, we can use it for data visualization. - When $r < k-1$, we lose information when we do the reduction.
Quadratic discriminant analysis (QDA)	Idea: Instead of LDA's assumption of having the covariance matrix being the same across groups, we allow each group to have its own covariance matrix Σ_k. Everything else is the same as LDA. Results in quadratic boundaries between the classes.	+ More flexible model than LDA, good when $n \gg p$. - Many more parameters to estimate than LDA.
Regularized discriminant analysis	Idea: When there is not enough data, we can regularize the covariance matrices. Mixture of QDA and LDA: Let $\widehat{\Sigma}_j(\alpha) = \alpha \widehat{\Sigma} + (1-\alpha)\widehat{\Sigma}_j$, where $\alpha \in [0,1]$ is a tuning parameter. Shrink towards identity covariance: $\widehat{\Sigma}(\alpha) = \alpha I \hat{\sigma}^2 + (1-\alpha)\widehat{\Sigma}$.	+ Good for situations where there is insufficient data to support LDA or QDA. + Good when the estimated covariance matrices have low rank.

Method	Description & Assumptions	Pros & Cons
Flexible discriminant analysis	Supervised learning: to construct a classifier. Idea: If we have k groups and n observations, construct the indicator matrix Y with $Y_{ij} = 1$ if observation i is in group j, 0 otherwise. Let $\theta_1, \ldots, \theta_L : \{1, \ldots, k\} \mapsto \mathbb{R}$ be $L \leq k-1$ scoring functions which are mean 0, variance 1 and orthogonal to each other. Then the solution to $\min_{\beta,\theta} \sum_{\ell=1}^{L} \sum_{i=1}^{n} [\theta_\ell(g_i) - x_i^T \beta_\ell]^2$ has $\beta_\ell \propto v_\ell$, the discriminant variables (defined on ESL p114). This allows us to generalize LDA in 2 ways: (a) use $f_\ell(x_i)$ in place of $x_i^T \beta_\ell$, and (b) add a penalty term to the minimization problem.	
Mixture discriminant analysis	Extension of LDA. Idea: Instead of assuming $X \mid$ in group $j \sim \mathcal{N}(\mu_j, \Sigma)$ for each group j, we assume $X \mid$ in group $j \sim$ mixture of normals with the same covariance matrix (both within the group and across groups). Model parameters can be estimated by the EM algorithm.	
Canonical correlation analysis (CCA)	Given 2 random vectors x and y, find a linear combination of entries of x and of y which maximize correlation with each other. More concretely, if $\Sigma_{11} = \mathbb{E}[(x-\mu)(x-\mu)^T]$, $\Sigma_{22} = \mathbb{E}[(y-\nu)(y-\nu)^T]$, $\Sigma_{12} = \mathbb{E}[(x-\mu)(y-\nu)^T]$, we want $\max_{a,b} a^T \Sigma_{12} b$ subject to $a^T \Sigma_{11} a = b^T \Sigma_{22} b = 1$. The solution is given by the SVD of $\Sigma_{12}^* = \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2}$. The constrained above can be modified to result in a and b being smooth.	
Classical metric scaling	Given a distance of dissimilarity matrix D, try to find points in $\mathbb{R}^k$ with distances given by D (or are close to it). Define $A_{ij} = -\frac{1}{2} D_{ij}^2$, $B = \left(I - \frac{11^T}{n}\right) A \left(I - \frac{11^T}{n}\right)$. Let $B = VDV^T$ be the eigendecomposition. Then rows of $Z = V_k D_k^{1/2} \in \mathbb{R}^{n \times k}$ is the solution. (Z solves $\underset{X}{\text{minimize}} \|B - XX^T\|_F$.)	+ Simple method for computing solution that has a closed form. + Has an inner product interpretation: it turns distances into inner products, then finds a low-dimensional embedding to approximate the inner product. It minimizes the strain $S_C(z_1, \ldots, z_n) = \sum_{i,i'} (s_{ii'} - \langle z_i - \bar{z}, z_{i'} - \bar{z} \rangle)^2$. - Assumes Euclidean distances.
Kruskal-Shepard metric scaling	Idea: Find a lower-dimensional representation of the data that preserves pairwise distances as well as possible, by minimizing stress function $S(z_1, \ldots, z_n) = \sum_{i \neq i'} (d_{ii} - |z_i - $	+ Works directly on distances, no need for inner product. - No closed form solution.

Method	**Description & Assumptions**	**Pros & Cons**
Kruskal-Shepard non-metric scaling	Idea: Distances D_{ij} may be far from Euclidean, but $f(D_{ij})$ may be closer for some monotone f. Seek to minimize stress function $S(z_1, \ldots, z_n) = \frac{\sum_{i \neq i'} (f(D_{ii'}) - \|z_i - z_{i'}\|)^2}{\sum_{i \neq i'} \|z_i - z_{i'}\|^2}$. Alternating solution: Given f, use gradient descent on stress to get z_i's. Given z_i's, find f by isotonic regression.	+ Works even if distances are far from Euclidean by using only ranks. - By the same token, only uses rank information, so potentially throwing away information.
Isometric feature mapping (ISOMAP)	Idea: Data actually lies on a manifold, so usual distances are misleading. Instead, use geodesic distances along the manifold. For each data point, find its neighbors (e.g. k nearest neighbors). Construct the neighborhood graph. Define geodesic distance between 2 points as the shortest path between them on this graph. Run classical metric scaling with these distances.	+ Works well when noise is small. - Computationally expensive. - Known to have difficulties for manifolds with "holes".
Local linear embedding (LLE)	Idea: Each point can be approximated by a linear combination of its neighbors. Construct a lower-dimensional set of points that preserves this relationship. For each x_i, find k nearest neighbors $\mathcal{N}(i)$. Approximate each point by a mixture of points in the neighborhood: $\min_w \|x_i - \sum_{k \in \mathcal{N}(i)} w_{ik} x_k\|^2$. Then, find points $y_1, \ldots y_n$ in lower-dimensional space to minimize $\sum_{i=1}^n \|y_i - \sum_{k \in \mathcal{N}(i)} w_{ik} y_k\|^2$. The solution turns out to be the trailing eigenvectors of $M = (I - W)^T (I - W)$ (ignoring the trivial eigenvector 1).	+ Preserves local structure well. + Less computationally expensive than ISOMAP. - Does not preserve global structure as well. - Known to have difficulty on non-convex manifolds.
Local MDS	Idea: Try to match local distances well; for points that are far apart, approximate distance by some large D (encourages them to be far apart). This is done by minimizing the local stress function $S(z_1, \ldots, z_n) = \sum_{(i,i') \in N} (d_{ii'} - \|z_i - z_{i'}\|)^2 + \sum_{(i,i') \notin N} w \cdot (D - \|z_i - z_{i'}\|)^2$, where N is the set of pairs of points which are considered close. For the problem to scale well, we need $w \sim 1/D$ as $D \to \infty$. When this happens, we have $S(z_1, \ldots, z_n) = \sum_{(i,i') \in N} (d_{ii'} - \|z_i - z_{i'}\|)^2 + \tau \sum_{(i,i') \notin N} \|z_i - z_{i'}\|$, where $\tau = 2wD$.	+ Most straightforward (compared to ISOMAP and LLE).

Method	**Description & Assumptions**	**Pros & Cons**
Spectral clustering	Start with a similarity/weight matrix $W \in \mathbb{R}^n$ (ones on the diagonal). Let G be a diagonal matrix with G_{ii} = sum of weights of edges connected to i. Find the m eigenvectors $Z \in \mathbb{R}^{n \times m}$ corresponding to the smallest eigenvalues of $\tilde{L} = I - G^{-1}W$. Then apply an unsupervised learning procedure (e.g. k-means clustering) to the rows of Z. Idea: For unnormalized $L = G - W$, we can show that $\frac{1}{2}\sum_{i,i'} w_{ii'}(f_i - f_{i'})^2 = f^T L f$. If we think of f_i as a score for observation i, then we want $(f_i - f_{i'})^2$ to be small when $w_{ii'}$ is large. This amounts to minimizing $f^T L f$.	+ Good for finding non-convex clusters. - We have to choose the measure of similarity with its associated parameters, the number of eigenvectors of $\tilde{L}$ and any parameters for the final clustering step.