Dimensionality Reduction Algorithms for Medical Imaging: PCA, NMF, ICA, etc.

Ben Kandel

July 13, 2012

Introduction

Dimensionality reduction algorithms can be written as matrix factorization problems. The basic form of the factorization is something like

$$\mathbf{X} \approx \mathbf{W}\mathbf{V}^{\mathrm{T}} \tag{1}$$

Books and articles have different notations for the dimensionality of $\mathbf{X}$ and the different matrices (so sometimes you will see that $\mathbf{X}$ is the data matrix, and sometimes $\mathbf{X}^{\mathrm{T}}$), and the transposes are propagated along all the steps of the decomposition. We'll try to be consistent and use the following notation:

- The data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ has n rows, with each row corresponding to an observation (one subject), and p columns, with each column correponding to a variable (voxel).
- The loading matrix $\mathbf{W} \in \mathbb{R}^{n \times r}$ corresponds to the reduced-dimensionality version of $\mathbf{X}$, replacing the p variables with $r \ll p$ variables.
- The basis matrix $\mathbf{V} \in \mathbb{R}^{p \times r}$ corresponds to the eigenvector matrix from classical PCA.

Using this notation, the loading matrix $\mathbf{W}$ corresponds to the projections of the data matrix on the basis matrix, $\mathbf{W} = \mathbf{X}\mathbf{V}$.

PCA

The most widely used dimensionality reduction algorithm is PCA. PCA finds an orthogonal rotation of the covariance matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ that satisfies one of the following problems:

1. Maximize variance in projected space:

$$\begin{aligned} \underset{\mathbf{V}}{\text{maximize}} \quad & \|\mathbf{X}\mathbf{V}\|_2^2 \\ \text{subject to} \quad & \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I} \end{aligned} \tag{2}$$

2. Minimizing reconstruction error:

$$\begin{aligned} \underset{\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{X}\|_2^2 \\ \text{subject to} \quad & \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I} \end{aligned} \tag{3}$$

In either case, orthogonality in the projected space is enforced.

Sparse PCA

In sparse PCA, additional constraints are enforced on the eigenvectors of the covariance matrix. This gives us something like

$$\underset{\mathbf{V}}{\text{minimize}} \|\mathbf{X} - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{X}\|_2^2 + \lambda \sum_i |V_i|, \tag{4}$$

where each column in $\mathbf{V}$ is V_i. Enforcing sparsity, though, normally entails discarding orthogonality. Because the "eigenvectors" are not orthogonal, one component of the data matrix may project onto more than one "eigenvector," so total variance explained is not an appropriate measure of how good an approximation to the original matrix the sparse eigenvectors are. There are several different ways of dealing with this issue. One way that seems reasonable to me is that proposed by Shen [7], which computes the "adjusted variance explained" as the difference between the projections of the data matrix projected on the column space of the rank-k and the rank-(k-1) sparse approximations of the data matrix. This variance explained is then normalized by the norm of the data matrix so that it scales from 0 to 1 to obtain the percentage of explained variance. In equations, the adjusted variance explained of the k'th component is given by $\mathrm{tr}\left(\mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k\right) - \mathrm{tr}\left(\mathbf{X}_{k-1}^{\mathrm{T}}\mathbf{X}_{k-1}\right)$, with $\mathbf{X}_k = \mathbf{X}\mathbf{V}_k\left(\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k\right)^{-1}\mathbf{V}_k^{\mathrm{T}}$ and $\mathbf{V}_k$ the first k vectors of the sparse basis matrix.

Most versions of SPCA still do have some sort of constraints, though:

1. Changing notation to standard SVD notation ($\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$), [8] enforces $\|\mathbf{u}\|_2^2 \leq 1$, $\|\mathbf{v}\|_2^2 \leq 1$ (but not orthogonality). This is sort of a lower bound on the kinds of restraints you have–each vector must at the very least not increase the size (i.e., ℓ_2 norm) of the original matrix.

2. Jolliffe [5] *does* enforce orthogonality on the eigenvectors. The trade-off this presents is that the output is not uncorrelated: $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$, but $\mathbf{V}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{V}$ is not diagonal. Although Jolliffe does not explicitly address why he chooses orthogonality over uncorrelatedness, it seems that most authors who choose to maximize variance enforce orthogonality. The reason this seems to be the case is that without a reconstruction error term, maximizing variance without orthogonality may just return the best rank-1 approximation of the data matrix over and over again. When incorporating reconstruction error into the decomposition, incorporation of orthogonality is not as necessary (`http://ai.stanford.edu/~quocle/LeKarpenkoNgiamNg.pdf`).

3. Zou [9] splits the eigenvector matrices ($\mathbf{V}\mathbf{V}^{\mathrm{T}}$) in two, $\mathbf{A}\mathbf{B}^{\mathrm{T}}$. He then enforces orthogonality on $\mathbf{A}$ and sparsity on $\mathbf{B}$:

$$\underset{\mathbf{A},\mathbf{B}}{\text{minimize}} \|\mathbf{X} - \mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{X}\|_2^2 + p(\mathbf{B}), \tag{5}$$

where $p(\cdot)$ is the sparsity penalty and $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}$.

4. Shen [7] only really gives results for a rank-one approximation of the data matrix. He basically iterates between projecting on $\mathbf{u}$ and $\mathbf{v}$, applying a thresholding operation, and re-scaling so that the vectors have unit norm. Except for the re-scaling (so that $\|\mathbf{v}\|_2^2 = 1$), there is no other constraint.

NMF

NMF has two objective functions (see "Algorithms for Non-Negative Matrix Factorization" in NIPS 2001):

1.
$$\begin{aligned} \underset{\mathbf{V},\mathbf{W}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{W}^{\mathrm{T}}\|_2^2 \\ \text{subject to} \quad & \mathbf{V}, \mathbf{W} \succeq 0 \end{aligned} \tag{6}$$

2.
$$\begin{aligned} \underset{\mathbf{V},\mathbf{W}}{\text{minimize}} \quad & D\left(\mathbf{X} \| \mathbf{V}\mathbf{W}^{\mathrm{T}}\right) \\ \text{subject to} \quad & \mathbf{V}, \mathbf{W} \succeq 0, \end{aligned} \tag{7}$$

where $D(X\|Y)$ is the divergence (KL, if they're probabilities) of the two arguments, having the form $D(A\|B) = \sum_{ij}\left(A_{ij}\log\frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right)$. No motivation is given for the introduction of the divergence metric as opposed to Frobenius norms. It's unclear to me what benefit it offers.

There are no orthogonality, diagonalization, etc. constraints at all in the original formulation. Since the original formulation, there have been a few additions that included different constraints. Li [6] includes an orthonormality constraint. He includes a soft penalty rather than a hard constraint, so that his objective function is of the form

$$\begin{aligned} \underset{\mathbf{V},\mathbf{W}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{W}^{\mathrm{T}}\|_2^2 + \lambda \sum_i \sum_{j \neq i} v_{ij} \\ \text{subject to} \quad & \mathbf{V}, \mathbf{W} \succeq 0 \end{aligned} \tag{8}$$

It seems to me that the reason that orthogonality is not necessary for NMF when reconstrution cost is used is that when non-negativity constraints are used, reconstruction costs take the place of orthogonality (like what we saw before for PCA, but even stronger, because there can't be cancellation). (This should probably be formalized.)

ICA

As opposed to PCA and variants, which aim to perform dimensionality reduction, ICA (independent components analysis) was originally proposed to solve the "blind source separation" problem in which the observed signals are a mixture of the source signals, which are statistically independent. Because of this focus of the problem, the nomenclature is different in ICA. The standard ICA nomenclature is as follows:

$$\mathbf{X} = \mathbf{A}\mathbf{s}, \tag{9}$$

where $\mathbf{X}$ is the observed data matrix, $\mathbf{A}$ is the "mixing" matrix, and $\mathbf{s}$ is the source matrix. In the nomenclature we are using, $\mathbf{A}$ corresponds to the loading matrix $\mathbf{W}$, and $\mathbf{s}$ corresponds to the basis matrix $\mathbf{V}$. In the ICA literature, the inverse of the mixing matrix $\mathbf{A}$ is denoted $\mathbf{W}$, but we will not be using that here to avoid confusion with the loading matrix. Using our notation, the parallel problem to the PCA objective function (Equation 3) is something like:

$$\begin{aligned} \underset{\mathbf{W},\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{W}\mathbf{V}\|_2^2 \\ \text{subject to} \quad & \text{Columns of } \mathbf{V} \text{ "independent."} \end{aligned} \tag{10}$$

In practice, though (at least as usually formulated), ICA uses a fundamentally different objective function from PCA. As explained before, PCA maximizes the variance of the projections onto the eigenvectors. This objective has a simple closed-form solution, but suffers from the drawback of assuming that the variance of the projections is a sufficient statistic to characterize the data. If the data follows a Gaussian distribution, the variance is a sufficient statistic. ICA, on the other hand, does not assume Gaussianity of the data. Because of this, it cannot use variance as a statistic. Instead, it first whitens the data so that it has unit variance, and then computes "eigenvectors" that give independent projections [2, 3, 4]. The reason for whitening is that the distribution of Gaussian variables is invariant under orthogonal transformation [2]. Because of this, using the variance of the dataset, which corresponds to second-order statistics, is only correct up to an orthogonal transformation. Once the data has been whitened, second-order statistics are no longer useful for further transformations, and higher-order statistics are necessary. Originally, kurtosis was used [1], but this has fallen out of favor and has been replaced by more robust approximations of non-Gaussianity.

Therefore, instead of using the variance of the projections, ICA (at least as implemented in the FastICA algorithm, which is the most widely used algorithm for ICA [3, 4]) uses the negentropy, which is a measure of how far apart a given random variable is from Gaussianity:

$$J_G(\mathbf{v}) = \left[E\left\{G\left(\mathbf{v}^{\mathrm{T}}\mathbf{x}\right)\right\} - E\left\{G(\nu)\right\}\right]^2, \tag{11}$$

where $\mathbf{v}$ is a column of $\mathbf{V}$ and $\mathbf{x}$ is a row of $\mathbf{X}$, ν is a Gaussian rv with unit variance (same as $\mathbf{v}$) and G is some non-quadratic "contrast" function, such as $\log\cosh(x)$.

$$\begin{aligned} \underset{\mathbf{v}}{\text{maximize}} \quad & \sum_{i=1}^{n} J_G\left(\mathbf{v_i}\right) \\ \text{subject to} \quad & \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I} \end{aligned} \tag{12}$$

Note that in this part of the decomposition, there is no reconstruction cost term. Because there is no dimensionality reduction at all in this stage and the transformation can be full rank, it is invertible and the original data can be recovered. The reason for the rotation here is to produce highly kurtotic sources, not to reconstruct the data.

Connection between SPCA and ICA

This is intended as a sketch of the connection between SPCA and ICA. If ICA is done in a one-step way, in which both orthogonality and kurtosis of the vectors are imposed (so that the PCA whitening stage and the ICA rotation stage are performed at the same time), it will end up being very similar to SPCA. This is because the kurtotic penalty on the ICA vectors makes the entries in the ICA basis vectors have a non-Gaussian distribution–spikier, with a heavier tail. In practice, this corresponds to having many entries in the ICA basis vector be 0, and many have strongly non-0 values, the same idea as happens in an ℓ_1-based sparseness penalty. The reason for this can be seen clearly when we formulate the objective function:

$$\begin{aligned} \underset{\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{X}\| - (\log\cosh\mathbf{V} - \log\cosh\nu) \\ \text{subject to} \quad & \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I} \end{aligned} \tag{13}$$

The $\log\cosh$ function is very similar to an ℓ_1 penalty. It's not clear to me what exactly would change if we forgot about the negentropy step (difference between the contrast function applied to the basis vector and a Gaussian vector) and just applied the contrast function straight to the basis vector, but if we did, that would be straight SPCA. I think that this question is at the heart of the difference between SPCA and ICA, and can help us formulate them both in a common framework.

References

[1] Jean-Francois Cardoso. Source separation using higher order moments. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 4, pages 2109–2112, 1989.

[2] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[3] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 10(3):626 –634, May 1999.

[4] A. Hyvrinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[5] I.T. Jolliffe and M. Uddin. The simplified component technique: An alternative to rotated principal components. *Journal of Computational and Graphical Statistics*, 9(4):689–710, 2000.

[6] S.Z. Li, X.W. Hou, H.J. Zhang, and Q.S. Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I207–I212, 2001.

[7] H. Shen and J.Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[8] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics (Oxford, England)*, 10(3):515–534, July 2009. PMID: 19377034.

[9] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265—286, 2006.