Linked NMF for Signal Source Separation

Learning shared and unique feature models across sample sets with implicitly linked factorizations

Zach DeBruine

2021-10-28

NMF problem definition

Basic NMF

Non-negative matrix factorization (NMF) enforces non-negativity in place of orthogonality to learn an additive, low-rank representation of some non-negative matrix $A_{N \times M}$ by minimizing the Frobenius norm:

$$\min_{\{W,H\} \geq 0} \|A - WH\|_F^2 \tag{1}$$

where $(W)_{N \times k}(H)_{k \times M}$ of rank k produce a lower-rank approximation of A.

Generally, W is randomly initialized and H and then W are alternately updated until some stopping criteria is satisfied, such as a maximum number of iterations or a measure of convergence.

Joint NMF (jNMF)

Joint NMF integrates multiple datasets with a common set of observations. For K data matrices $(A_1)_{N \times M_1}, ..., (A_K)_{N \times M_K}$, the objective is:

$$\min_{\{W,H\} \geq 0} \sum_{k=1}^{K} \|A_k - WH_k\|_F^2 \tag{2}$$

Notice that eqn. 2 on separate datasets is equivalent to eqn. 1 on a combined dataset since each dataset contributes to the loss function equally.

jNMF cannot separate shared and unique signals between datasets because there is only one W matrix mapped to each dataset A_k by a single H_k matrix.

Integrative NMF (iNMF)

Integrative NMF, proposed by Yang and Michailidis, can resolve shared and unique signals between datasets, subject to linear and additive correspondences between these signals.

iNMF considers shared signals in WH_k and unique signals in U_kH_k. The following is a perspective on iNMF:

$$\min_{\{W,H,U\}\geq 0} \sum_{k=1}^{K} \| A_k - (W + \lambda U_k) H_k] \|_F^2 \tag{3}$$

To retain identifiability of shared signals, the contribution of unique signals ($U_k H_k$) to the model of shared signals ($W H_k$) is weighted by λ.

iNMF assumes direct correspondence between shared and unique effects because W and U_k are added and mapped to A_k by the same weights in H_k. Thus, W gives the minimum additive basis of shared signal in $A_{1 \ldots K}$ while $U_{1 \ldots K}$ gives additional unique signal, and thus each factor in W and U_k contain linearly coordinated information.

This can be a limitation, for example, in separation of male- and female-specific gene expression where W should describe non-specific processes and U_{male} or U_{female} should describe sex-specific processes, in which case iNMF would improperly assume linear coordination and additivity between sex-specific and non-specific processes.

Linked NMF (lNMF)

U_k may be uncoupled from W by introducing a unique mapping matrix for unique signal, V_k. This approach relaxes the assumptions of linear and additive correspondence between W and U_k in iNMF.

The result is that each dataset is described by unique effects in $U_k V_k$ and shared effects in $W H_k$, such that $A_k \approx W H_k + U_k V_k$. In other words, unique factorizations are "linked" by the shared model, W.

A useful perspective of such "linked NMF" (lNMF) is the following:

$$\min_{\{W,H,U,V\}\geq 0} \sum_{k=1}^{K} \left\| A_k - \begin{pmatrix} W & U_k \end{pmatrix} \begin{pmatrix} H_k \\ V_k \end{pmatrix} \right\|_F^2 \tag{4}$$

In lNMF, factors in the unique signal model (U_k) need not coordinate with factors in the shared signal model (W). Furthermore, the complexity of U_k may differ from W because rank may be varied.

In principle, lNMF is an extension of jNMF in which unique factors in U (mapped by V) are concatenated to shared factors in W (mapped by H).

Unlike in iNMF, there is no need for a weighting parameter (λ) to retain identifiability because W and U_k are mapped jointly, and the relative ranks of U_k and W control the resolution of unique and shared signals.

The following perspective of lNMF illustrates the separability of the two linked matrix factorization subproblems:

$$\min_{\{W,H,U,V\}\geq 0} \sum_{k=1}^{K} \| A_k - (W H_k + U_k V_k) \|_F^2 \tag{5}$$

A separation of the two objectives in this expression makes clear that linked NMF implicitly links two factorization problems:

$$\min_{\{W,H,U,V\}\geq 0} \sum_{k=1}^{K} \| A_k - W H_k \|_F^2 + \sum_{k=1}^{K} \| A_k - U_k V_k \|_F^2 \tag{5}$$

Obviously the models must be jointly considered during each update, and thus the above perspective is not particularly useful for deriving solutions.

Linked NMF in Transfer Learning

Transfer learning (TL) by linear model projection minimizes the expression:

$$\min_{H_0 \geq 0} \|A_0 - WH_0\|_F^2 \tag{6}$$

In the above expression, W has been trained on some data A and is now being projected onto some new data A_0 to find H_0. In other words, H_0 is the mapping matrix for W onto A_0.

However, this objective does not alternately refine W and H_0, and thus the minimization of the objective is entirely dependent on how well the available information in W explains the information in A_0. Thus, if A and A_0 contain non-overlapping signal, W cannot ideally minimize the TL objective.

Most transfer learning projections are degenerate, because W is not an exhaustive dictionary of all signal that may possibly be encountered in A_0. The mapping in H_0 involves sub-optimal and possibly entirely incorrect mapping, which may mislead interpretation of the results.

As a solution to this problem, consider a linked TL objective:

$$\min_{H_0, U, V \geq 0} \left\| A_0 - \begin{pmatrix} W & U \end{pmatrix} \begin{pmatrix} H_0 \\ V \end{pmatrix} \right\|_F^2 \tag{7}$$

Here, TL involves projection of W onto new data A_0 to find H_0 alongside additional factors in UV that explain additional signal in A_0 not in A.

The rank of U must be decided based on a tradeoff point that balances error of the model against mapping preference for W over U.

Solving NMF problems

NMF is commonly solved using multiplicative updates, as proposed by Seung and Lee, or some type of block-coordinate pivoting method, such as alternating least squares (ALS) updates. ALS, subject to non-negativity constraints, has become popular due to its definite convergence guarantee and performance.

Solving NMF with Alternating Least Squares

To solve the NMF problem in eqn. 1, W is randomly initialized, and H and then W are alternately updated until some stopping criteria is satisfied. The alternating updates are column-wise in H and row-wise in W:

$$H_{:i} \leftarrow \min_{H_{:i} \geq 0} \|A_{:i} - WH_{:i}\|_F^2 \tag{8}$$

$$W_{:j}^T \leftarrow \min_{W_{:j}^T \geq 0} \|A_{:j}^T - H^T W_{:j}^T\|_F^2 \tag{9}$$

$$\forall ij, \ where \ 1 \leq i \leq N, 1 \leq j \leq M$$

One way to minimize this expression with non-negative least squares (NNLS) is to find an equivalent form as $ax = b$, derived from eqn. 8, where a is symmetric positive definite:

$$W^T W H_{:i} = W^T A_{:i} \quad \forall i, \ 1 \leq i \leq N \tag{10}$$

where $a = W^TW$, $x = H_{:i}$, and $b = W^TA_{:i}$. W^TW is constant for all columns in H, thus the calculation of $W^TA_{:i}$ and NNLS solving may be massively parallelized.

The corresponding form for eqn. 7 is:

$$HH^TW_{:j}^T = HA_{:j}^T \quad \forall j,\ 1 \leq j \leq M \tag{11}$$

Algorithms for solving non-negative least squares (NNLS) are not discussed here.

Solving lNMF problems

In lNMF, shared and unique signals must be jointly resolved according to eqn. 4. Thus, each alternating update in lNMF consists of two minimization problems, one which is unique for each dataset A_k (i.e. the updates of H_k, U_k, and V_k), and one which is linked across all datasets $A_{1...K}$ (i.e. the update of W), where each problem must account for the current solution of the other.

Prior to updating, randomly initialize W, $U_{1...K}$, and $V_{1...K}$. $H_{1...K}$ may be uninitialized since they will be updated first.

Unique Updates

Solve the unique minimization problem in eqn. 4 to update H_k, V_k, and U_k.

The update for H_k and V_k as one unit, corresponding to eqn. 10, is the following:

$$\begin{pmatrix} W^T \\ U_k^T \end{pmatrix} \begin{pmatrix} W & U_k \end{pmatrix} \begin{pmatrix} H_{k:i} \\ V_{k:i} \end{pmatrix} = \begin{pmatrix} W^T \\ U_k^T \end{pmatrix} A_{:i} \quad \forall i,\ 1 \leq i \leq N_k \tag{12}$$

where W, U_k, and A are fixed. Let $Y_k = \begin{pmatrix} W & U_k \end{pmatrix}$, then realize that $a = Y^TY$, $b = Y^TA_{:i}$ and $x = \begin{pmatrix} H_{k:i} \\ V_{k:i} \end{pmatrix}$. Note that H_k and V_k are resolved simultaneously.

The corresponding update for U_k, corresponding to eqn. 9, is the following:

$$\begin{pmatrix} H_k \\ V_k \end{pmatrix} \begin{pmatrix} H_k^T & V_k^T \end{pmatrix} \begin{pmatrix} W_{:j}^T \\ U_{k:j}^T \end{pmatrix} = \begin{pmatrix} H_k \\ V_k \end{pmatrix} A_{:j}^T \quad \forall j,\ 1 \leq j \leq M \tag{13}$$

where W, A, H_k, and V_k are fixed. Let $Z = \begin{pmatrix} H_k \\ V_k \end{pmatrix}$, then realize that $a = ZZ^T$, $b = ZA_{:j}^T$, and $x = \begin{pmatrix} W_{:j}^T \\ U_{k:j}^T \end{pmatrix}$.

In eqn. 13, x is partially fixed in $W_{:j}^T$. It is important to hold $W_{:j}^T$ constant in this case, and not to even partially update it, since the contributions of $H_{1...K \notin k}$ would confound the update. The update of $W_{:j}^T$ is thus necessarily "linked" across all datasets $A_{1...K}$.

Linked Updates

Update W by minimizing the loss of the linked factorizations while holding U, V, and H constant for all datasets $A_{1...K}$:

$$\min_{W \geq 0} \left\| \begin{pmatrix} A_1^T \\ \vdots \\ A_K^T \end{pmatrix} - \begin{pmatrix} H_1^T & V_1^T & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ H_K^T & 0 & 0 & V_K^T \end{pmatrix} \begin{pmatrix} W^T \\ U_1^T \\ \vdots \\ U_K^T \end{pmatrix} \right\|_F^2 \tag{14}$$

For brevity, consider eqn. 14 to be in the form:

$$\min_{W \geq 0} \left\| A^T - \begin{pmatrix} H^T & tr(V^T_{1...K}) \end{pmatrix} \begin{pmatrix} W^T \\ U^T \end{pmatrix} \right\|_F^2 \tag{15}$$

where $tr(V^T_{1...K})$ is the diagonal matrix spelled out in eqn. 14 and H^T and U^T are marginal concatenations of $H_{1...K}$ and $U_{1...K}$.

Let $X = \begin{pmatrix} H^T & tr(V^T_{1...K}) \end{pmatrix}$, the update of W corresponding to eqn. 11 is thus:

$$XX^T \begin{pmatrix} W^T_{:j} \\ U^T_{:j} \end{pmatrix} = XA^T_{:j} \quad \forall j,\ 1 \leq j \leq M \tag{16}$$

where $U^T_{:j}$ is fixed, similarly to what was the case for $W^T_{:j}$ in eqn. 13. Realize that $a = XX^T$, $b = XA^T_{:j}$, and $x = \begin{pmatrix} W^T_{:j} \\ U^T_{:j} \end{pmatrix}$.

Adapting canonical NMF updating algorithms for lNMF

A much simpler approach for updating WU and HV, which comes with very little computational penalty, is to consider linked NMF as an $A = WH$ factorization problem, where $W = \begin{pmatrix} W & U_1 & \cdots & U_K \end{pmatrix}$ and $H = X$, in which zeros in H are maintained with each update. Thus, initial W gives a random initialization while initial H gives the linking matrix. With each update of H, b is only computed from rows in W^T that are non-zero in rows of H_i.

This approach is implemented in RcppML, and comes with very little computational penalty despite the much more elegant updating procedure.

Determination of Ranks

In lNMF, the shared signal factorization $WH_{1...K}$ is of a certain rank, while each unique signal factorization U_kV_k is of a different rank. Each rank is necessarily at least 1. The true difficulty of the rank determination problem thus scales exponentially with the number of datasets, K.

As a near-exact (and incredibly expensive) approach to the problem of rank-determination, suppose all datasets are factorized jointly (jNMF) at a rank that minimizes some cross-validation objective. This gives the largest possible rank (D_0) for $WH_{1...K}$ in a linked factorization, if all signal were shared. Now suppose all datasets are factorized individually (NMF) at a rank that minimizes some cross-validation objective. This gives the largest possible rank ($D_{1...K}$) for U_kV_k in a linked factorization, if all signal were unique. Evidently, these ranks form the outer boundaries of possible scenarios, since if any signal is shared between datasets, these ranks will be overestimates. Thus, a theoretically exact method for determining optimal ranks would involve an iterative rank-downdating procedure involving alternate rank-reduction of WH followed by each of the U_kV_k models to points each that minimizes some cross-validation objective.

As an approximate (and generally satisfactory) approach to the problem of rank-determination, suppose all datasets ($A_{1...K}$) are factorized independently (NMF) to a rank (D_K) that minimizes some cross-validation objective. Now let the rank of the $WH_{1...K}$ model in lNMF be set to the number of factors conserved across all independent factorizations (D_0) as determined by some similarity heuristic. Let the rank of each U_kV_k model be set to $D_k - D_0$, and at least 1. This approach requires only a single lNMF run, and is reasonably approximate.

Extending lNMF

Because lNMF relies on updates by alternating least squares, it can take advantage of functionality supported by basic NMF algorithms. This includes massive parallelization, masking, L1/L2 regularization, and diagonalization.

Linked NMF implementation

Linked NMF is implemented in the Rcpp Machine Learning Library (RcppML) R package, version 0.5.2 or greater.

```
# devtools::install_github("zdebruine/RcppML")
library(RcppML)
```

The `RcppML::lnmf` function takes a list of datasets, a rank for the shared matrix (`k_wh`), ranks for each of the unique matrix (`k_uv`), and parameters also used in the `nmf` implementation.

The example below uses the `aml` dataset to find common signal between AML and reference cell methylation signatures.

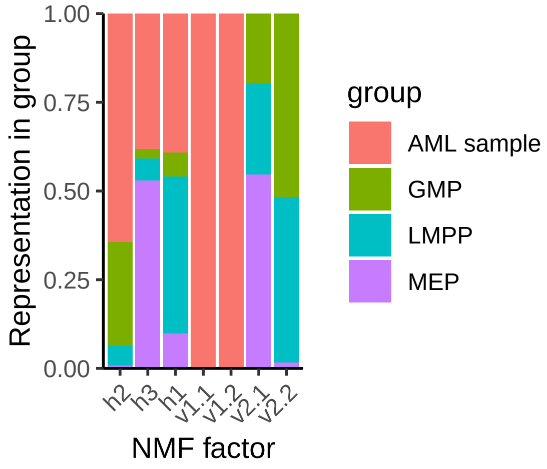
```
data <- list(
  aml[, which(colnames(aml) == "AML sample")],
  aml[, which(colnames(aml) != "AML sample")]
)

lnmf_model <- lnmf(data, k_wh = 3, k_uv = c(2, 2))
```

Convert the `lnmf` model to an `nmf` model and plot factor representation in each sample grouping:

```
nmf_model <- as(lnmf_model, "nmf")

library(ggplot2)
plot(summary(nmf_model,
             group_by = colnames(aml),
             stat = "mean"))
```



As expected, lNMF has generated 3 factors describing shared signal (h1-3), two factors describing signal specific to AML samples (v1.1-2), and two factors describing signal specific to reference cell types (v2.1-2). In this case, these results are useful in classifying AML samples based on from which of the three reference cell types they likely originate.

References

Material in this markdown was heavily inspired by work from Yang and Michailidis and the Welch lab. NMF code for demonstration purposes is derived from the RcppML package and the NMF implementation described in DeBruine et al. 2021.