Comp726

Lecture 4

October 29,2025

Today

- Finish discussion on data leakage
- Naive bayes as a generative classification model
- One strategy for multiomics data integration through subspace merging

Data leakage continued

- This is the most common mistake → if you have multiple samples from the same donor, you cannot have some of their instances in the training set and some of their instances in the test set.
- Example : let's say we are profiling immune cells in blood to predict whether donors were responders or non responders to some treatment. Let's say we get samples from each donor at multiple timepoints.
- If we were to split all of the samples randomly into train and test sets, then there would be instances from the same donor in both the training and test sets. This means that we have already seen information about 'test donors' in training.
- What to do instead → Split donors between training and test sets. Keep all instances from the same donor together.

Figure: Example: donors with many sample instances.

One last data leakage pitfall : double-dipping with feature selection

- Feature selection is another common pre-processing technique that is helpful for interpretability and model accuracy (more on this later)
- Feature selection should be done only using the training set
- **Less dire example**: Selecting features with high variance across data points should technically be done on the training set
- **Very dire example:** Let's say you applied logistic regression to the entire dataset, took only the features with high magnitude coefficients and trained a model only with that information. This would have double-dipped in the dataset.

Common issue: too many features

- This is not really a pitfall, as much as it is an artifact of working with modern biological datasets.
- If the number of measured features (p) is significantly larger than the number of profiled samples, N, then it is very easy for the model to overfit the training data. There is some combination of features that could explain each data point.
- The common solution is to **regularize** the loss function with a penalty term, which forces many coefficients to be 0.
- Lasso, ridge, and elastic net are common penalization approaches that are used.
- For example: The lasso penalty augments the loss function, $\mathcal{L}$ as $\mathcal{L} + \lambda \sum_{j=1}^{p} |\beta_j|$

Quick overview of a *generative* classifier: naive bayes

- **objective:** We seek to infer $P(X \mid Y)$, so, the probability of a class label, given measured features. Our $X = [X_1, X_2, \ldots, X_n]$ is our collection of measured features. For now, let's assume that each X_i is one of discrete values.
- **assumption:** Each features, X_i, is conditionally independent of the Xs as $P(X_1, \ldots, X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$
- How can we predict a discrete class label, Y given our measured features (specifically belonging to class y_k)?

Deriving naive bayes

$$P(Y = y_k \mid X_1 \ldots X_n) = \frac{P(Y = y_k)\, P(X_1 \ldots X_n \mid Y = y_k)}{\sum_j P(Y = y_j)\, P(X_1 \ldots X_n \mid Y = y_j)} \tag{1}$$

Then from our conditional independence assumption,

$$P(Y = y_k \mid X_1 \ldots X_n) = \frac{P(Y = y_k) \prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i \mid Y = y_j)} \tag{2}$$

The probability of each class is then ...

$$Y \leftarrow \arg\max_{y_k} \frac{P(Y = y_k) \prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i \mid Y = y_j)} \tag{3}$$

Training :

$$\hat{P}(X_i = x_i \mid Y = y) = \frac{(\ \#\ \text{training examples where } X_i = x_i \text{ and } Y = y)}{(\ \text{training examples where } Y = y)} \tag{4}$$

and,

$$\hat{P}(Y = y) = \frac{(\#\ \text{training examples where } Y = y\ )}{(\#\ \text{training examples})} \tag{5}$$

Using naive bayes to classify a cell as microglia or not

Training

Iba1	Cd11b	CD68	CD3	microglia
high	high	high	low	yes
high	high	low	low	yes
low	low	low	high	no
low	low	high	high	no
high	low	high	low	yes

Testing example

Iba1	Cd11b	CD68	CD3	microglia
high	high	high	low	

Figure: we can use the matrix of training examples to classify a new example by estimating combinations of X and Y.

Unsupervised analysis of multiomics data

Classical Omics Integration Problem

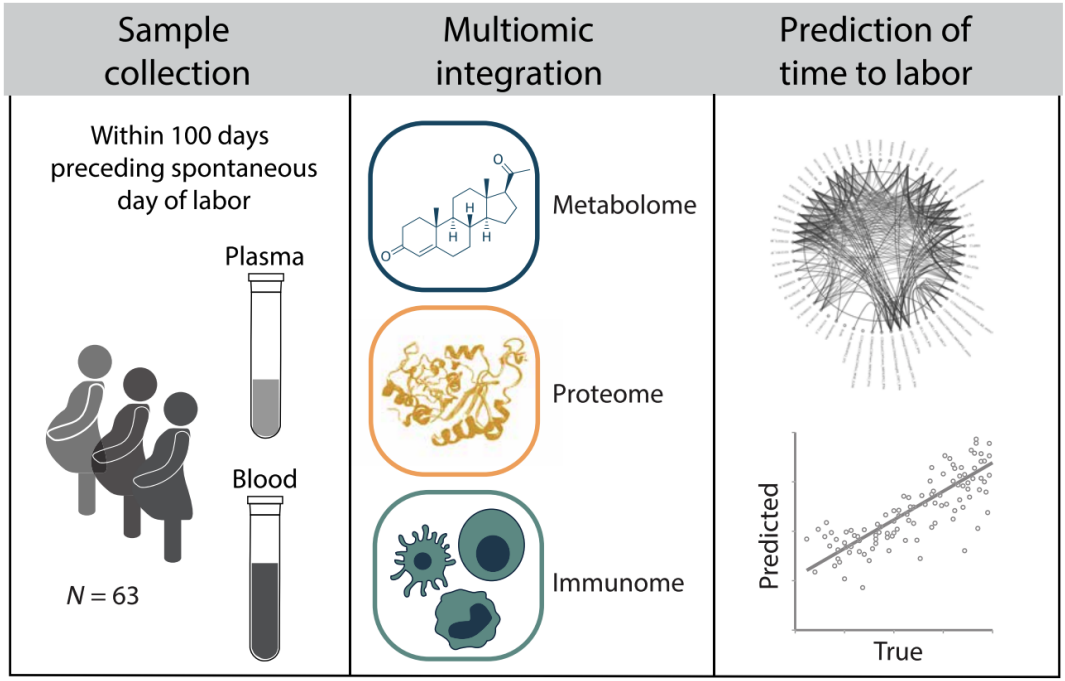

Figure: Figure from Stelzer *et al.* Science Translational Medicine. 2021. How do we leverage disparate modalities to predict something about patients, given inherent properties and quirks of each dataset?

The Cancer Genome Atlas (TCGA)

The cancer Genome Atlas was one of the first major profiling efforts, collecting diverse types of data across many patients, cancers, and biological modalities.

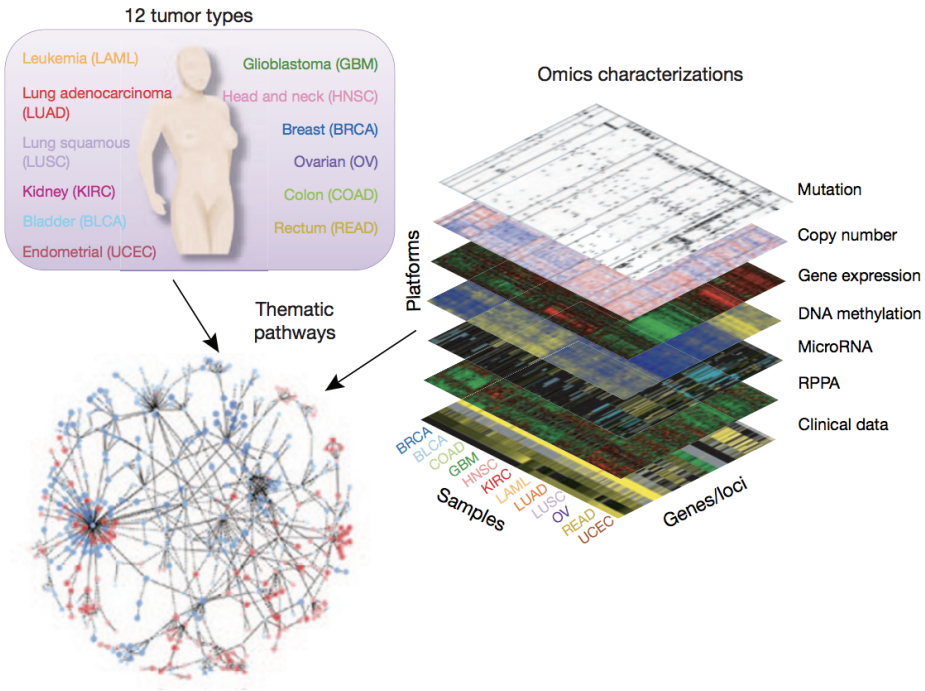

Figure: from TCGA, Nature Genetics. 2013.

Notation and Problem Formulation

- Consider M types of omics data measurements $\{\mathbf{X}^m\}_{m=1}^M$ from the same set of N patients.
- For a modality, m, there are p_m measured features and the dimensions of the data matrix are therefore $p_m \times N$
- We will let G^m be the graph for modality m
- **Goal**: We seek a joint subspace embedding, $\mathbf{U} \in \mathbb{R}^{N \times k}$ that is representative of all modalities.

Overview of Subspace Merging

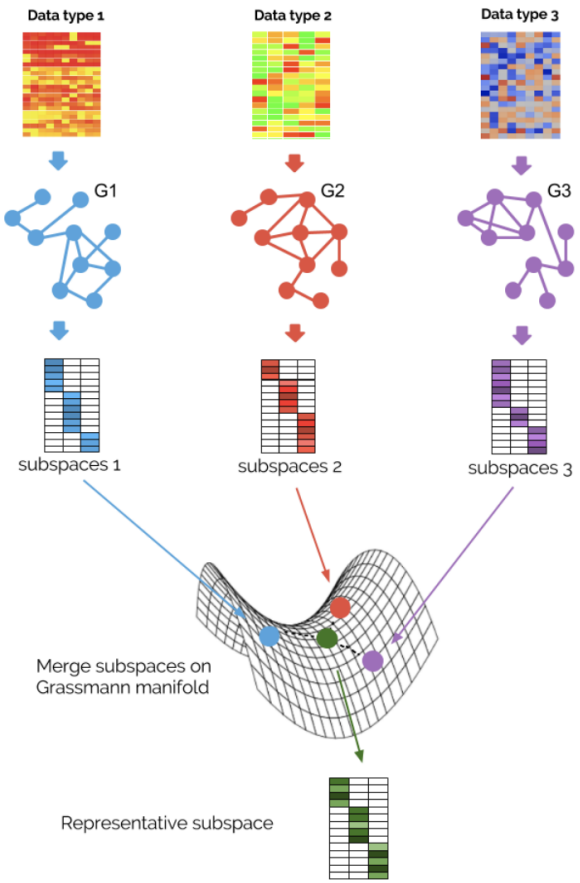

Figure: from Ding *et al.* Bioinformatics. 2019.

What is a Grassmann Manifold?

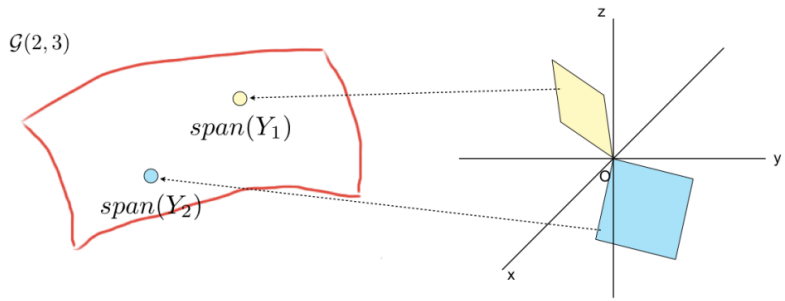

Figure: Example of $\mathcal{G}(2,3)$

- Notation, $\mathcal{G}(k,n)$ is the set of k-dimensional linear subspaces in $\mathbb{R}^n$, such that each subspace is a point.
- So, each point on $\mathcal{G}$ can be represented by an orthonormal matrix $\mathbf{Y} \in \mathbb{R}^{n \times k}$
- **Selling Point:** We know how to talk about how geometrically close the subspaces are, based on principal angles

Why is this useful to our problem?

- **Each Modality Graph As A Subspace:** From each modality, we create a graph. We can ultimately compute the joint subspace or *embedding* given individual subspaces.
- **Well-Defined Distances Measures:** We know how to compute distances between subspaces on the Grassmannian. The representative subspace, $\mathbf{U}$ should be equidistant from the per-modality subspaces ($\mathbf{U}^m$s).

Build a Similarity Graph Between Patients in Each Modality

Calculate edge weights as,

$$S_{ij}^{m} = \exp\left(-\frac{\left\|\mathbf{x}_i^m - \mathbf{x}_j^m\right\|^2}{2t^2}\right), i = 1, \ldots, N, j = 1, \ldots .N$$

From here, retain the top k edges for each node based on S_{ij} and use W_{ij} for the notation of the edge weights retained, such that, $W_{ij}^m = S_{ij}^m$

Quadratic form helps with optimization problem

We can compute the total variation of some signal, $\mathbf{x}$ across the graph, based on a summary of the the graph called the *graph laplacian*.

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}(x_i - x_j)^2 \tag{6}$$

This is useful for us, because you can think of $\mathbf{x}$ as a dimension of the embedding coordinates, which should not vary too much across the graph[1].

[1] $\mathbf{A}$ is an adjacency matrix for the graph so that $A_{ij} = 1$ if node i is connected to node j and is 0 otherwise

Pause for Rayleigh Ritz Theorem

Let $\mathbf{A}$ be a square, symmetric matrix, $N \times N$ matrix with eigenvalues, $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ and corresponding eigenvectors [2] $\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_n\}$. Then define

$$R_{\mathbf{A}}(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}. \tag{7}$$

Then the minimum value of $R_{\mathbf{A}}(\mathbf{x})$ is λ_1 and it's taken for $\mathbf{x} = \mathbf{v}_1$

[2] you can think of eigenvectors as the basis of a matrix, which encodes unique signal in a matrix

Matrix Extension

We will be seeing a lot of the form of $\mathbf{X}^T\mathbf{L}\mathbf{X}$. We can talk about the trace of that matrix product as the distance in vectors of adjacent nodes.

$$\text{trace}(\mathbf{X}^T\mathbf{L}\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}||\mathbf{x}_i - \mathbf{x}_j|| \tag{8}$$

An extension of Rayleigh Ritz says that the minimum k-dimensional matrix $\mathbf{X}$ of trace$(\mathbf{X}^T\mathbf{L}\mathbf{X})$ is $\lambda_1 + \lambda_2. + \cdots + \lambda_k$ and is obtained using the first k eigenvectors of $\mathbf{L}$, as $\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_k\}$.

Specify Optimization Problem in terms of Normalized Graph Laplacian

$$\mathbf{L}^m = \mathbf{D}^{m^{-\frac{1}{2}}} \left(\mathbf{D}^m - \mathbf{W}^m\right) \mathbf{D}^{m^{-\frac{1}{2}}}$$

Written out this gives us,

$$L_{i,j}^{\text{sym}} := \begin{cases} 1 & \text{if } i = j \text{ and } \deg(v_i) \neq 0 \\ -\frac{1}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}$$

Writing Down the Objective Function

The goal is to specify a $\mathbf{U}^m$ for each modality. The optimal graph embedding in k dimensions can written as,

$$\min_{\mathbf{U}^m \in \mathbb{R}^{N \times k}} \operatorname{tr}\left(\mathbf{U}^{m'} \mathbf{L}^m \mathbf{U}^m\right), \quad \text{s.t. } \mathbf{U}^{m\prime} \mathbf{U}^m = I$$

- It turns out the solution is the first k eigenvectors of the Graph Laplacian $\mathbf{L}^m$ by the Rayleigh–Ritz theorem

3

[3] Note that the $U^{m'}$ refers to the transpose of $\mathbf{U}^m$

Defining a Projection Distance Between The Integrative Subspace and Individual Modality Subspaces

$$
\begin{aligned}
d^2_{\text{proj}}\left(\mathbf{U}, \{\mathbf{U}^m\}_{m=1}^M\right) &= \sum_{m=1}^{M} d^2_{\text{proj}}\left(\mathbf{U}, \mathbf{U}^m\right) \\
&= \sum_{m=1}^{M} \left[k - \text{tr}\left(\mathbf{U}\mathbf{U}'\mathbf{U}^m\mathbf{U}^{m\prime}\right)\right] \\
&= kM - \sum_{i=1}^{M} \text{tr}\left(\mathbf{U}\mathbf{U}'\mathbf{U}^m\mathbf{U}^{m\prime}\right)
\end{aligned}
$$

The subspace, $\mathbf{U}$ that minimizes this is close to all individual subspaces, $\{\mathbf{U}^m\}_{i=1}^M$

Optimization Problem for Multiple Subspaces

The optimization problem for merging multiple subspaces finally can be written as,

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times k}} \sum_{m=1}^{M} \text{tr}\left(\mathbf{U}'\mathbf{L}^m\mathbf{U}\right) + \alpha\left[kM - \sum_{m=1}^{M} \text{tr}\left(\mathbf{U}\mathbf{U}'\mathbf{U}^m\mathbf{U}^{m\prime}\right)\right], \quad \text{s.t. } \mathbf{U}'\mathbf{U} = I$$

It can be shown that this simplifies to,

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times k}} \text{tr}\left[\mathbf{U}'\left(\sum_{i=1}^{M}\mathbf{L}^m - \alpha\sum_{m=1}^{M}\mathbf{U}^m\mathbf{U}^{m\prime}\right)\mathbf{U}\right], \quad \text{s.t. } \mathbf{U}'\mathbf{U} = I$$

Rayleigh Ritz Again....

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times k}} \text{tr}\left[\mathbf{U}'\left(\sum_{i=1}^{M}\mathbf{L}^m - \alpha\sum_{m=1}^{M}\mathbf{U}^m\mathbf{U}^{m\prime}\right)\mathbf{U}\right], \quad \text{s.t. } \mathbf{U}'\mathbf{U} = I$$

Hopefully you recognize the form of the objective. We can define a new matrix, $\mathbf{L}_{\text{mod}}$ and again the first k eigenvectors are the optimal solution. Or,

$$\mathbf{L}_{mod} = \sum_{m=1}^{M}\mathbf{L}^m - \alpha\sum_{m=1}^{M}\mathbf{U}^m\mathbf{U}^{m\prime}$$

Recap

1. **Per-Modality Graph:** Create a graph for each modality and compute corresponding Laplacians. These form the points on the Grassmannian.
2. **Quadratic Form for Per-Modality Subspaces :** In general we want to ensure each subspace dimension respects each modality's graph structure and hence yields a small value for the quadratic form.
3. **Projection Distance:** The ultimate joint subspace, $\mathbf{U}$ should be as close as possible to per-modality subspaces, $\mathbf{U}^m s$
4. **Apply Rayleigh Ritz:** Objective is formulated in a way that we know the optimal solution is the first k eigenvectors.

Clustering on Merged Subspace

When you cluster on the merged subspace, you get groups with different prognostic interpretations.

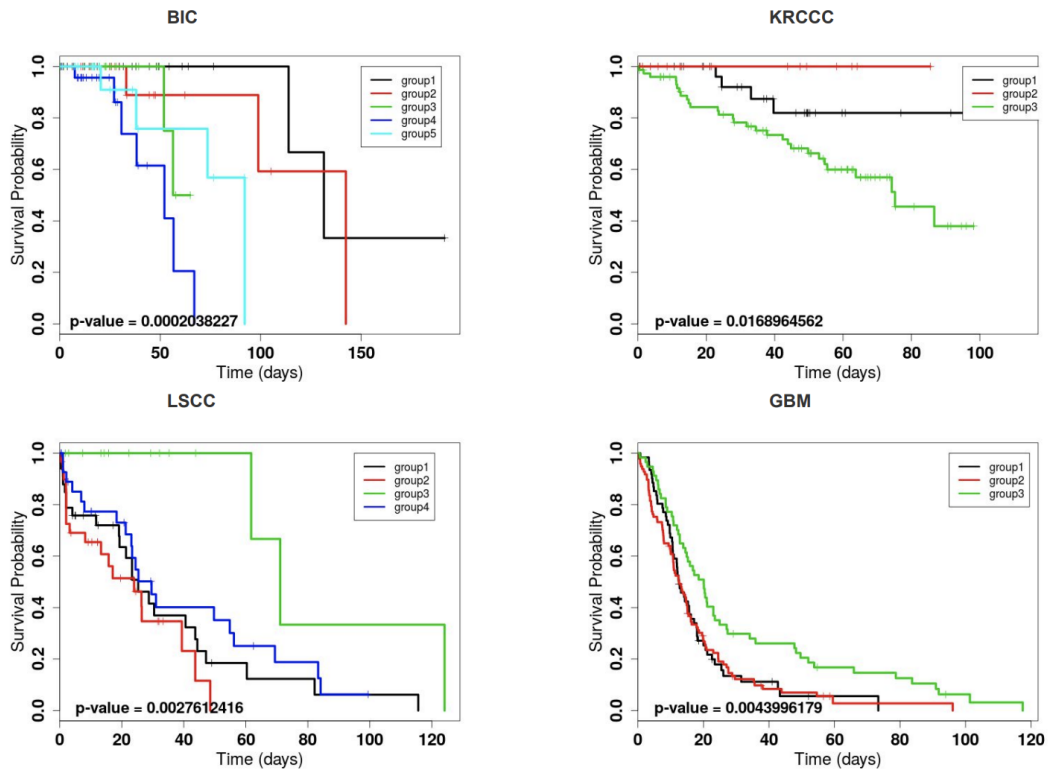

Figure: from Ding *et al.* Bioinformatics. 2018.