MLFPM 2021 Summer School

Machine learning techniques for data integration

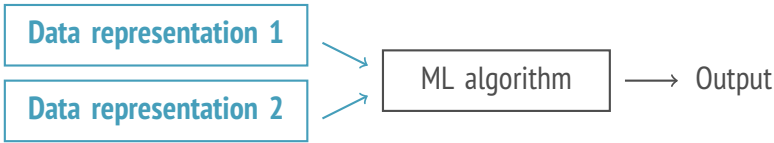
Chloé-Agathe Azencott

Center for Computational Biology (CBIO)
Mines ParisTech – Institut Curie – INSERM U900
PSL Research University & PR[AI]RIE, Paris, France

September 20, 2021

http://cazencott.info chloe-agathe.azencott@mines-paristech.fr @cazencott

Data integration/fusion



- Several **views** of the data (or **modalities**)

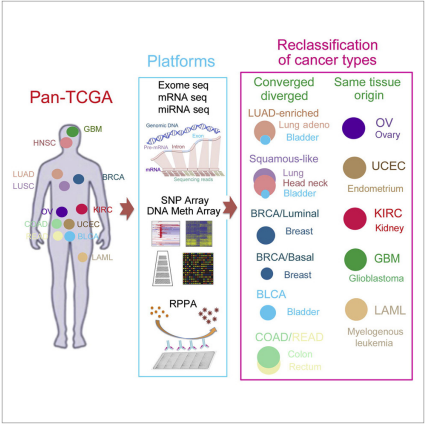
$$\begin{cases} \{\vec{x}_1^1, \vec{x}_2^1, \ldots, \vec{x}_n^1\}, \vec{x}_i^1 \in \mathcal{X}^1 \\ \{\vec{x}_1^2, \vec{x}_2^2, \ldots, \vec{x}_n^2\}, \vec{x}_i^2 \in \mathcal{X}^2 \\ \ldots \end{cases}$$

- **multi-view** (or **multi-modal**) ML: **learning** jointly from these views
- **Assumption:** the views are **complementary**

Examples of multi-view learning problems

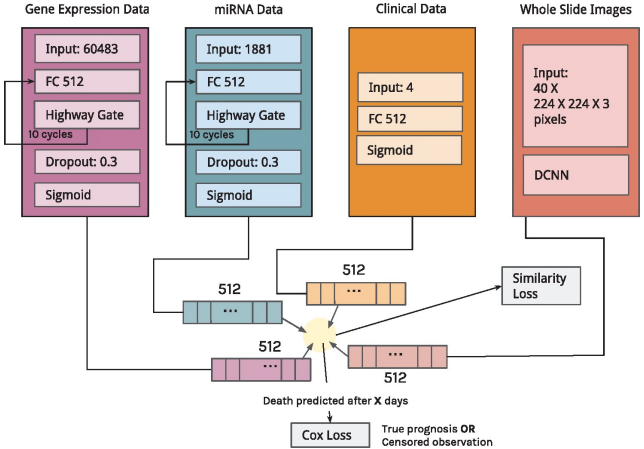
Disease subtyping from multiomics data

Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification within and across Tissues of Origin [Hoa+14]



Multimodal prognosis prediction

- **Multimodal pancancer prognosis prediction** [CG19]
- Combine multiomics and images



Stage of integration

Early integration

– Concatenate the features $\longrightarrow$ single-view problem

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } f : \mathbb{R}^{p_1+p_2} \to \mathcal{Y}$$

Early integration

- Concatenate the features $\longrightarrow$ single-view problem

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } f : \mathbb{R}^{p_1+p_2} \to \mathcal{Y}$$

- ☺ Easy to set up

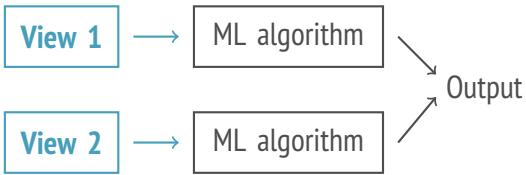
Early integration

- Concatenate the features $\longrightarrow$ single-view problem

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } f : \mathbb{R}^{p_1+p_2} \to \mathcal{Y}$$

- ☺ Easy to set up
- ☹ Combining apples and oranges: **normalization** and **interpretability?**
- ☹ **Curse of dimensionality** $\longrightarrow$ learning is difficult

Late integration

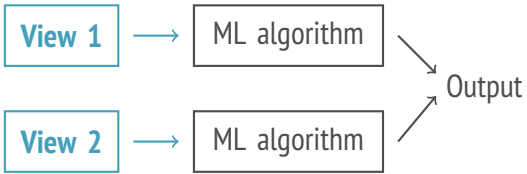


- Combine the outputs of the different views

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } \begin{cases} f_1 : \mathbb{R}^{p_1} \to \mathcal{Y} \\ f_2 : \mathbb{R}^{p_2} \to \mathcal{Y} \end{cases} \quad \text{and } g : \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}.$$

- g can be **learned** or **pre-set** (e.g. majority vote)

Late integration

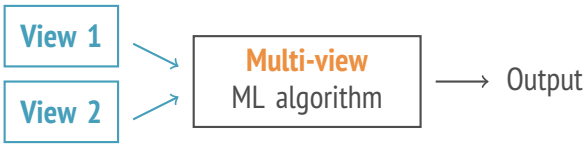


- Combine the outputs of the different views

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } \begin{cases} f_1 : \mathbb{R}^{p_1} \to \mathcal{Y} \\ f_2 : \mathbb{R}^{p_2} \to \mathcal{Y} \end{cases} \quad \text{and } g : \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}.$$

- g can be **learned** or **pre-set** (e.g. majority vote)
- ☺ Easy to set up
- ☹ Ensemble learning works better if models are **uncorrelated** → hard to benefit from more than one view

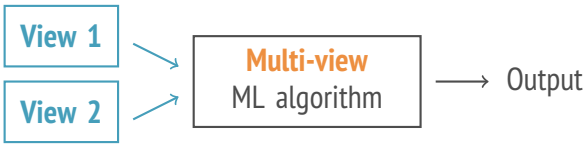
Intermediate integration



– Jointly transform the data at the same time as learning

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } f : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathcal{Y}$$

Intermediate integration

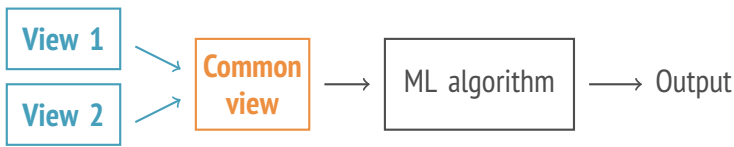


- Jointly transform the data at the same time as learning

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn } f : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathcal{Y}$$

- ☺ Explicitely models multiplicity of data sources
- ☹ Requires new algorithms

Idea 1: Embed the data in a common feature space



Idea 1: Embed the data in a common feature space

1. Find a **low-dimensional** representation of the data

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn} \quad \varphi : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathbb{R}^p \text{ with } p \ll p_1 + p_2$$

Examples:

- Joint non-negative matrix factorization
- Deep multi-view learning

Joint non-negative Matrix Factorization (NMF)

- **NMF** for a single view $X \in \mathbb{R}^{n \times p}$

  Find $W \in \mathbb{R}_+^{n \times d}$, $H \in \mathbb{R}_+^{d \times p}$ s.t. $X \approx WH$

$$\min_{W,H} ||X - WH||_F^2$$

- **Joint NMF** for V views $X^1 \in \mathbb{R}^{n \times p_1}, \ldots, X^V \in \mathbb{R}^{n \times p_V}$

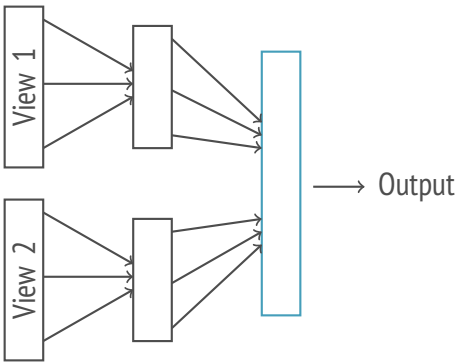
$$\min_{W,H^1,\ldots,H^V} \sum_{v=1}^{V} \theta_v ||X^v - WH^v||_F^2$$

- For more, see [Can+21] and A. Baudot's talk

[LS01; Zha+12; CF17]

Deep multiview learning

- The last layer of a (deep) neural network is a **learned shared representation** of all views
- Can be **supervised** (feed forward) or **unsupervised** (autoencoder)



[Gao+20]

Idea 1: Embed the data in a common feature space

1. Find a **low-dimensional** representation of the data

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn} \quad \varphi : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathbb{R}^p \text{ with } p \ll p_1 + p_2$$

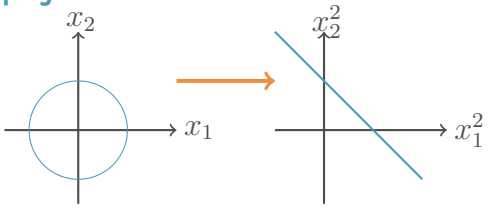
2. Map the data to a **high-dimensional** Hilbert space

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn} \quad k : \mathbb{R}^{p_1+p_2} \times \mathbb{R}^{p_1+p_2} \to \mathbb{R}$$

Kernel methods

- Build **non-linear** models as linear models over a **mapping of the data** to a (higher dimensional, Hilbert) space

  E.g. $\varphi_{\text{quad}} : (x_1, x_2, \ldots, x_p) \mapsto (x_1, x_2, \ldots, x_p, x_1^2, x_1x_2, \ldots, x_p^2)$



- **Kernels** are **dot products** in Hilbert spaces

$$k(\vec{x}, \vec{x}') = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle_{\mathcal{H}} \qquad k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \qquad \varphi : \mathcal{X} \to \mathcal{H}$$

- **Kernel trick:** any algorithm where inputs $\vec{x}$ only appear in dot products can be **kernelized** by replacing $\langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle$ with $k(\vec{x}, \vec{x}')$

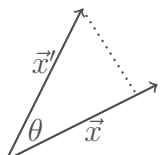
  E.g. $k_{\text{quad}} : \vec{x}, \vec{x}' \mapsto (\langle \vec{x}, \vec{x}' \rangle + 1)^2$

- Useful if computing k is easier than computing φ

Kernel methods

- Kernels can be interpreted as measures of **similarity**
- Examples
  - **RBF Gaussian** kernel: $k(\vec{x}, \vec{x}') = \exp\left(-\frac{||\vec{x}-\vec{x}'||_2^2}{2\sigma^2}\right)$
  - **string matching** kernels for protein sequences
  - **diffusion** kernels based on interaction networks
  - **graph** kernels for molecules
  - **identical-by-state** kernels for blocks of SNPs

[Sch+04; Bor+20]



$$\langle \vec{x}, \vec{x}' \rangle = ||\vec{x}||\ ||\vec{x}'||\cos\theta$$

Multiple kernel learning

- One kernel per view: $K^1, K^2, \ldots, K^V \qquad K^v \in \mathbb{R}^{n \times n} \qquad K^v_{il} = k_v(\vec{x}_i, \vec{x}_l)$
- **multi-view kernel:** $K = \sum_{v=1}^{V} \mu_v K^v$

  There exists a Hilbert space $\mathcal{H}$ and a mapping $\varphi : (\mathcal{X}^1 \times \cdots \times \mathcal{X}^V) \to \mathcal{H}$ such that $K((\vec{x}^1, \ldots, \vec{x}^V), (\vec{x}^{1\prime}, \ldots, \vec{x}^{V\prime})) = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle_{\mathcal{H}}$
- **Support Vector Machine** (dual formulation)

$$\max_{\vec{\alpha} \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{l=1}^{n} \alpha_i \alpha_l y_i y_l k(\vec{x}_i, \vec{x}_l)$$

- The smaller this max, the better the performance $\rightarrow$ **optimize the kernel while learning**

$$\min_{\mu_1, \ldots, \mu_v} \max_{\vec{\alpha} \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{l=1}^{n} \alpha_i \alpha_l y_i y_l \sum_{v=1}^{V} \mu_v k_v(\vec{x}_i, \vec{x}_l)$$

[Lan+04; DB+07]

Idea 2: Force the models learned on each view to match

Idea 2: Force the models learned on each view to match

- Learn **one model per view,** but constrain the models to **agree**

$$\text{From } \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{learn} \quad \begin{cases} f_1 : \mathbb{R}^{p_1} \to \mathcal{Y} \\ f_2 : \mathbb{R}^{p_2} \to \mathcal{Y} \end{cases} \quad \text{s.t. } f_1 \approx f_2$$

- Final model = simple **combination of the individual models** (average, majority vote)
- ☺ Possible to make predictions even if **one view is missing**

Examples:

- Canonical correlation analysis
- **Regularization**
  - SVM-2K
  - Multi-view Rank Minimization Lasso
  - Multi-NMF

Canonical Correlation Analysis (CCA)

- Find **basis vectors** so as to **maximize correlation** between the projections of two views

$$\max_{\vec{w}^1,\vec{w}^2} \vec{w}^{1\top} X^{1\top} X^2 \vec{w}^2 \quad \text{s.t.} \quad \vec{w}^{1\top} X^{1\top} X^1 \vec{w}^1 = \vec{w}^{2\top} X^{2\top} X^2 \vec{w}^2 = 1$$

- Equivalently, **minimize disagreement** between the projections

$$\min_{\vec{w}^1,\vec{w}^2} ||X^1\vec{w}^1 - X^2\vec{w}^2||_2^2 \quad \text{s.t.} \quad ||X^1\vec{w}^1||_2^2 = ||X^2\vec{w}^2||_2^2 = 1$$

- Extends to V views

$$\min_{\vec{w}^1,\ldots\vec{w}=V} \sum_{v=1}^{V} \sum_{t>v} ||X^v\vec{w}^v - X^t\vec{w}^t||_2^2 \quad \text{s.t.} \quad ||X^v\vec{w}^v||_2^2 = 1 \text{ for } v = 1,\ldots,V$$

- Many extensions, including using kernels

[Hot36; Ket71; Yam+03; HSST04; WT09]

Regularization

- View-specific models are learned by solving an optimization problem

  E.g. (regularized) empirical risk minimization

$$f_v \in \arg\min_{f_v \in \mathcal{F}_v} \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, f_v(\vec{x}_i^v))}_{\text{loss/risk}} + \lambda_v \underbrace{\Omega(f_v)}_{\text{regularization}}$$

  - Ridge regression: $\mathcal{F}_v = \{\vec{x} \mapsto \langle \vec{\beta}, \vec{x} \rangle\}$ $\quad L(y, f(\vec{x})) = (y - f(\vec{x}))^2$ $\quad \Omega(f) = \left|\left|\vec{\beta}\right|\right|_2^2$
  - SVM: $\mathcal{F}_v = \{\vec{x} \mapsto \sum_{i=1}^{n} \alpha_i y_i k(\vec{x}_i, \vec{x})\}$ $\quad L(y, f(\vec{x})) = \max(0, 1 - y f(\vec{x}))$ $\quad \Omega(f) = \left|\left|\vec{\beta}\right|\right|_2^2$

- Use a regularizer to tie the models together

$$\arg\min_{f_1, \ldots, f_V} \sum_{v=1}^{V} \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, f_v(\vec{x}_i^v))}_{\text{loss/risk}} + \lambda_v \underbrace{\Omega(f_v)}_{\text{regularization}} + \lambda\, \Omega_{\text{consensus}}(f_1, \ldots, f_v)$$

SVM-2K

- **Support Vector Machine** (primal formulation)

$$\min_{\vec{w},b} \frac{1}{2} ||\vec{w}||_2^2 + C \sum_{i=1}^{n} \xi_i \quad \text{s.t. } y_i \left(\langle \vec{w}, \vec{x}_i \rangle + b\right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \quad [i = 1, \ldots, n]$$

- **SVM-2K**

$$\min_{\vec{w}^1, \vec{w}^2, b_1, b_2} \underbrace{\frac{1}{2} \left|\left|\vec{w}^1\right|\right|_2^2 + C_1 \sum_{i=1}^{n} \xi_i^1}_{\text{view 1}} + \underbrace{\frac{1}{2} \left|\left|\vec{w}^2\right|\right|_2^2 + C_2 \sum_{i=1}^{n} \xi_i^2}_{\text{view 2}} + \underbrace{D \sum_{i=1}^{n} \eta_i}_{\text{consensus}}$$

s.t., for $i = 1, \ldots, n$, $\left|\left(\langle \vec{w}^1, \vec{x}_i^1 \rangle + b_1\right) - \left(\langle \vec{w}^2, \vec{x}_i^1 \rangle + b_2\right)\right| \leq \eta_i + \epsilon, \quad \eta_i \geq 0,$

$y_i \left(\langle \vec{w}^1, \vec{x}_i^1 \rangle + b_1\right) \geq 1 - \xi_i^1, \quad \xi_i^1 \geq 0, \quad y_i \left(\langle \vec{w}^2, \vec{x}_i^2 \rangle + b_2\right) \geq 1 - \xi_i^2, \quad \xi_i^2 \geq 0$

Prediction: $f((\vec{x}^1, \vec{x}^2)) = \frac{1}{2} \left(\langle \vec{w}^1, \vec{x}^1 \rangle + b_1 + \langle \vec{w}^2, \vec{x}^2 \rangle + b_2\right)$

[Far+06]

Multi-view Rank Minimization Lasso (MRM-Lasso)

- Single-view **Lasso**

$$\min_{\vec{w}} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle \vec{w}, \vec{x}_i \rangle)^2 + \lambda \, ||\vec{w}||_1$$

- **MRM-Lasso**
  - Also learn a low-rank matrix $M \in \mathbb{R}^{n \times V}$ such that M_{iv} quantifies the contribution of view v to label i.

$$\min_{\vec{w}^1, \ldots, \vec{w}^V, M} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \sum_{v=1}^{V} M_{iv} \langle \vec{w}^v, \vec{x}_i^v \rangle \right)^2 + \sum_{v=1}^{V} \lambda \, ||\vec{w}^v||_1 + \text{Rank}(M)$$

- Prediction: $f((\vec{x}^1, \ldots, \vec{x}^V)) = \sum_{v=1}^{V} m_v \langle \vec{w}^v, \vec{x}^v \rangle$ with $m_v = \frac{\sum_{i=1}^{n} M_{iv}}{\sum t \neq v \sum_{i=1}^{n} M_{it}}$

[Yan+15]

Multi non-negative Matrix Factorization (NMF)

- **NMF** for a single view $X \in \mathbb{R}^{n \times p}$

  Find $W \in \mathbb{R}_+^{n \times d}$, $H \in \mathbb{R}_+^{d \times p}$ such that $X \approx WH$

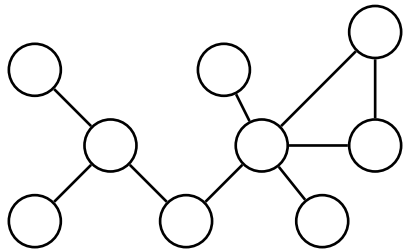
$$\min ||X - WH||_F^2$$

- **MultiNMF** for V views $X^1 \in \mathbb{R}^{n \times p_1}, \ldots, X^V \in \mathbb{R}^{n \times p_V}$

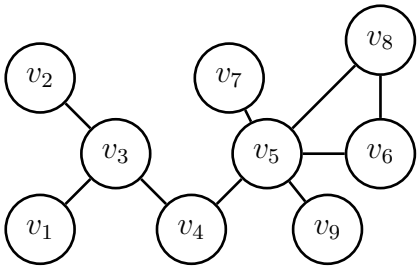
$$\min_{H^1,\ldots,H^V,W^1,\ldots,W^V,W} \sum_{v=1}^{V} ||X^v - W^v H^v||_F^2 + \lambda \sum_{v=1}^{V} ||W^v - W||_F^2$$

[Liu+13]

Idea 3: Use graphs

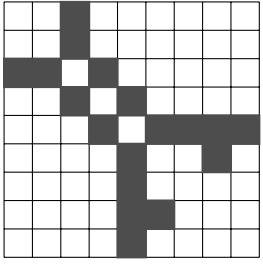


Graphs model relationships between entities



- **Graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V} = \{v_1, v_2, \ldots, v_9\}$
  - $\mathcal{E} = \{(v_1, v_3), (v_2, v_3), \ldots, (v_6, v_8)\}$
- **Adjacency matrix** $A \in \{0, 1\}^{9 \times 9}$

  $A_{il} \neq 0$ iff. $(v_i, v_l) \in \mathcal{E}$



- Variants:
  - Oriented edges: $(v_i, v_l) \neq (v_l, v_i)$
  - Weighted edges: $A_{il} = a_{il} \in \mathbb{R}_+$

Idea 3: Use graphs

1. Use graphs to represent **relationships between samples**

$$\text{Represent} \begin{cases} \{\vec{x}_i^1\}_{i=1,\dots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\dots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{as} \quad \begin{cases} A^1 \in \mathbb{R}^{n \times n} & \text{(adjacency matrix of a graph with } n \text{ vertices)} \\ A^2 \in \mathbb{R}^{n \times n} & \text{(adjacency matrix of a graph with } n \text{ vertices)} \end{cases}$$

$\rightarrow$ Learning as a **node labeling** problem (see A. Baudot)

Idea 3: Use graphs

1. Use graphs to represent **relationships between samples**

$$\text{Represent} \begin{cases} \{\vec{x}_i^1\}_{i=1,\ldots,n} \in \mathbb{R}^{p_1} \\ \{\vec{x}_i^2\}_{i=1,\ldots,n} \in \mathbb{R}^{p_2} \end{cases} \quad \text{as} \quad \begin{cases} A^1 \in \mathbb{R}^{n \times n} & \text{(adjacency matrix of a graph with } n \text{ vertices)} \\ A^2 \in \mathbb{R}^{n \times n} & \text{(adjacency matrix of a graph with } n \text{ vertices)} \end{cases}$$

2. Use graphs to **tie views together**

   When features from each view can be **mapped** to a vertex in a graph

   Examples:

   - Multi graph kernels
   - Graph-guided feature selection
   - Knowledge-informed neural network architecture

Mapping omics features to a graph

- Build graphs over genes from
  - **prior knowledge:** biological networks (pathways, coexpression, etc.)
  - **one view:** correlations between features
- Mapping omics features to genes:
  - Straightforward: mRNA transcripts; protein levels; somatic mutations
  - Single Nucleotide Polymorphisms: map to gene j all SNPs that are
    - in/near gene j **on the DNA sequence;**
    - known to **regulate** the expression of gene j;
    - in/near gene j **in 3D space** (Hi-C contact maps)
- One view = one set of **node labels**

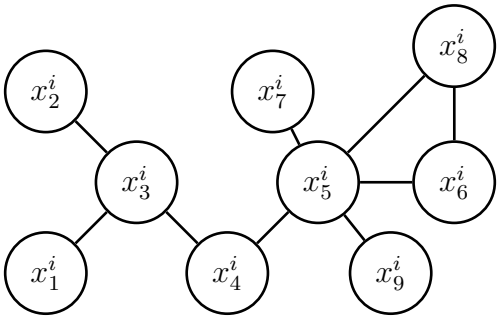
[Dur+20]

Graph-guided approaches

1. Use **graph kernels** to build one kernel per view; apply **multiple kernel** approaches.

   E.g. PAMOGK (Pathway Multi-Omics Graph Kernel based Approach) [Tep+20]

2. Use **graph regularization** to guide **feature selection** [Aze16]
3. Also see K. van Steen



Predictive models: graph-regularized Lassos

- Use a **lasso** (ℓ_1 regularizer) to encourage **sparsity**
- Use a **regularizer** that encourages connected features to have similar weights

$$\frac{1}{n}\sum_{i=1}^{n}\underbrace{L(y_i,\langle\vec{\beta},\vec{x}_i\rangle)}_{\text{loss/risk}}+\lambda_s\underbrace{\left|\left|\vec{\beta}\right|\right|_1}_{\text{sparsity}}+\lambda_g\ \Omega_{\text{graph}}(\vec{\beta})$$

- **Generalized fused Lasso:** $\Omega_{\text{graph}}(\vec{\beta})=\sum_{(j,k)\in\mathcal{E}}|\beta_j-\beta_k|$
- **Network-constrained Lasso:** $\Omega_{\text{graph}}(\vec{\beta})=\vec{\beta}^\top L\vec{\beta}=\frac{1}{2}\sum_{(j,k)\in\mathcal{E}}(\beta_j-\beta_k)^2A_{jk}$

$$L_{jk}=\begin{cases}\text{deg}(j)=\sum_k A_{jk} & \text{if } j=k\\ -A_{jk} & \text{otherwise}\end{cases}$$

[Tib+05; LL08]

Selection models: SConES & SigMod

- Compute the **relevance** r_j of each feature (e.g. association test score, correlation, HSIC)
- Select **few** features with **high relevance** and **connected** on the graph

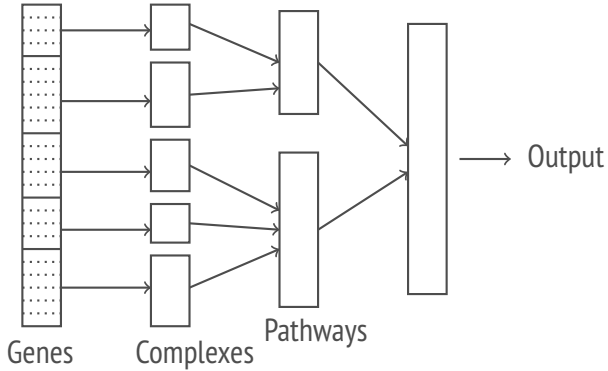
$$\max_{\mathcal{S} \subseteq \mathcal{V}} \sum_{j \in \mathcal{S}} r_j - \lambda_s |\mathcal{S}| - \lambda_g \Omega_{\text{graph}}(\mathcal{S})$$

- **SConES:** $\Omega_{\text{graph}}(\mathcal{S}) = \sum_{j \in \mathcal{S}} \sum_{k \notin \mathcal{S}} A_{jk}$ (penalizes disconnected solutions)
- **SigMod:** $\Omega_{\text{graph}}(\mathcal{S}) = -\sum_{j \in \mathcal{S}} \sum_{k \in \mathcal{S}} A_{jk}$ (encourages connected solutions)

[Aze+13; Liu+17; CGA21]

Knowledge-informed neural network architecture

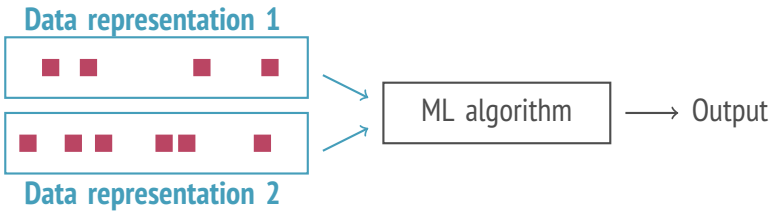
– Each hidden layer has a biological meaning; connect neurons accordingly



– See also the work of Pelin Gündoğdu & Joaquin Dopazo

[Yu+18; Ma+18; Hao+20; Gau+20]

Feature selection & intepretability



■ Important features

Feature selection & interpretability

- Apply **single-view techniques** to early or late integration
  - ☺ Easy to set up
  - ☹ Do not benefit from **joint** learning
- Sparse variants of multi-view algorithms
  - Sparse NMF [Hoy04; PJ+21]
  - Sparse CCA [WT09; LC+09]
  - MRM Lasso
  - Graph-guided feature selection
- **Attention** for deep learning

References I

[Aze+13] Chloé-Agathe Azencott et al. "Efficient network-guided multi-locus association mapping with graph cuts". en. In: *Bioinformatics* 29.13 (July 2013), pp. i171–i179. DOI: 10.1093/bioinformatics/btt238.

[Aze16] Chloé-Agathe Azencott. "Network-Guided Biomarker Discovery". In: *Machine Learning for Health Informatics*. Springer, 2016, pp. 319–336.

[Bor+20] Karsten Borgwardt et al. "Graph Kernels: State-of-the-Art and Future Challenges". English. In: *Foundations and Trends® in Machine Learning* 13.5-6 (Dec. 2020). Publisher: Now Publishers, Inc., pp. 531–712. DOI: 10.1561/2200000076.

[Can+21] Laura Cantini et al. "Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer". en. In: *Nature Communications* 12.1 (Jan. 2021), p. 124. DOI: 10.1038/s41467-020-20430-7.

[CF17] Prabhakar Chalise and Brooke L. Fridley. "Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm". eng. In: *PloS One* 12.5 (2017), e0176278. DOI: 10.1371/journal.pone.0176278.

[CG19] Anika Cheerla and Olivier Gevaert. "Deep learning with multimodal representation for pancancer prognosis prediction". In: *Bioinformatics* 35.14 (July 2019), pp. i446–i454. DOI: 10.1093/bioinformatics/btz342.

[CGA21] Héctor Climente-González and Chloé-Agathe Azencott. "martini: an R package for genome-wide association studies using SNP networks". In: *bioRxiv* (2021).

References II

[DB+07] Tijl De Bie et al. "Kernel-based data fusion for gene prioritization". In: *Bioinformatics* 23.13 (July 2007), pp. i125 –i132. DOI: 10.1093/bioinformatics/btm187.

[Dur+20] Diane Duroux et al. "Interpretable network-guided epistasis detection". In: *bioRxiv* (2020).

[Far+06] Jason D. R Farquhar et al. "Two view learning: SVM-2K, theory and practice". In: *Advances in Neural Information Processing ystems* (2006).

[Gao+20] Jing Gao et al. "A Survey on Deep Learning for Multimodal Data Fusion". In: *Neural Computation* 32.5 (May 2020), pp. 829–864. DOI: 10.1162/neco_a_01273.

[Gau+20] Thomas Gaudelet et al. "Unveiling new disease, pathway, and gene associations via multi-scale neural network". eng. In: *PloS One* 15.4 (2020), e0231059. DOI: 10.1371/journal.pone.0231059.

[Hao+20] Jie Hao et al. "PAGE-Net: Interpretable and Integrative Deep Learning for Survival Analysis Using Histopathological Images and Genomic Data". eng. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 25 (2020), pp. 355–366.

[Hoa+14] Katherine A. Hoadley et al. "Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification within and across Tissues of Origin". en. In: *Cell* 158.4 (Aug. 2014), pp. 929–944. DOI: 10.1016/j.cell.2014.06.049.

References III

[Hot36] Harold Hotelling. "Relations between two sets of variates". In: *Biometrika* 28.3-4 (Dec. 1936), pp. 321–377. DOI: 10.1093/biomet/28.3-4.321.

[Hoy04] Patrik O. Hoyer. "Non-negative Matrix Factorization with Sparseness Constraints". In: *Journal of Machine Learning Research* 5.Nov (2004), pp. 1457–1469.

[HSST04] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. "Canonical Correlation Analysis: An Overview with Application to Learning Methods". In: *Neural Computation* 16.12 (Dec. 2004), pp. 2639–2664. DOI: 10.1162/0899766042321814.

[Ket71] J. Kettenring. "Canonical analysis of several sets of variables". In: (1971). DOI: 10.1093/BIOMET/58.3.433.

[Lan+04] Gert R. G. Lanckriet et al. "A statistical framework for genomic data fusion". In: *Bioinformatics* 20.16 (Nov. 2004), pp. 2626–2635. DOI: 10.1093/bioinformatics/bth294.

[LC+09] K.-A. Lê Cao et al. "Sparse canonical methods for biological data integration: Application to a cross-platform study". English. In: *BMC Bioinformatics* 10 (2009). DOI: 10.1186/1471-2105-10-34.

[Liu+13] Jialu Liu et al. "Multi-View Clustering via Joint Nonnegative Matrix Factorization". en-US. In: May 2013.

[Liu+17] Yuanlong Liu et al. "SigMod: an exact and efficient method to identify a strongly interconnected disease-associated module in a gene network". In: *Bioinformatics* 33.10 (2017), pp. 1536–1544.

References IV

[LL08] Caiyan Li and Hongzhe Li. "Network-constrained regularization and variable selection for analysis of genomic data". en. In: *Bioinformatics* 24.9 (May 2008), pp. 1175–1182. DOI: 10.1093/bioinformatics/btn081.

[LS01] Daniel Lee and H. Sebastian Seung. "Algorithms for Non-negative Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Vol. 13. MIT Press, 2001.

[Ma+18] Jianzhu Ma et al. "Using deep learning to model the hierarchical structure and function of a cell". en. In: *Nature Methods* 15.4 (Apr. 2018), pp. 290–298. DOI: 10.1038/nmeth.4627.

[NW20] Nam D. Nguyen and Daifeng Wang. "Multiview learning for understanding functional multiomics". en. In: *PLOS Computational Biology* 16.4 (Apr. 2020). Publisher: Public Library of Science, e1007677. DOI: 10.1371/journal.pcbi.1007677.

[PJ+21] Morgane Pierre-Jean et al. "PIntMF: Penalized Integrative Matrix Factorization Method for Multi-Omics Data". In: *arXiv:2103.03184 [stat]* (Mar. 2021). arXiv: 2103.03184.

[Sch+04] Bernhard Schölkopf et al., eds. *Kernel Methods in Computational Biology*. en. Computational Molecular Biology. Cambridge, MA, USA: MIT Press, July 2004. ISBN: 978-0-262-19509-6.

[Tep+20] Yasin Ilkagan Tepeli et al. "PAMOGK: a pathway graph kernel-based multiomics approach for patient clustering". In: *Bioinformatics* 36.21 (Nov. 2020), pp. 5237–5246. DOI: 10.1093/bioinformatics/btaa655. (Visited on 09/20/2021).

References V

[Tib+05] Robert Tibshirani et al. "Sparsity and smoothness via the fused lasso". en. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1 (Feb. 2005), pp. 91–108. DOI: 10.1111/j.1467-9868.2005.00490.x.

[WT09] Daniela M Witten and Robert J. Tibshirani. "Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data". In: *Statistical Applications in Genetics and Molecular Biology* 8.1 (June 2009), p. 28. DOI: 10.2202/1544-6115.1470.

[Yam+03] Y. Yamanishi et al. "Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis". eng. In: *Bioinformatics (Oxford, England)* 19 Suppl 1 (2003), pp. i323–330. DOI: 10.1093/bioinformatics/btg1045.

[Yan+15] Wanqi Yang et al. "MRM-Lasso: A Sparse Multiview Feature Selection Method via Low-Rank Analysis". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.11 (Nov. 2015). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 2801–2815. DOI: 10.1109/TNNLS.2015.2396937.

[Yu+18] Michael K. Yu et al. "Visible Machine Learning for Biomedicine". en. In: *Cell* 173.7 (June 2018), pp. 1562–1565. DOI: 10.1016/j.cell.2018.05.056.

[Zha+12] Shihua Zhang et al. "Discovery of multi-dimensional modules by integrative analysis of cancer genomic data". In: *Nucleic Acids Research* 40.19 (Oct. 2012), pp. 9379–9391. DOI: 10.1093/nar/gks725.

References VI

[Zit+19] Marinka Zitnik et al. “Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities”. en. In: *Information Fusion* 50 (Oct. 2019), pp. 71–91. DOI: 10.1016/j.inffus.2018.09.012.