# Sparse modelling and Multiview Analysis

Rob Tibshirani
Stanford University

*"Statistically Speaking"*
November 30, 2022

# Outline

1. Sparse prediction models– Lasso and Elastic Net
2. Multi-view Analysis
   - A motivating example
   - Existing approaches: early and late fusion
3. Cooperative learning
   - General form of cooperative learning
   - Direct and One-at-a-time algorithms
   - Cooperative regularized linear regression
4. Real multiomics data example
5. Extensions and takeaways

# Outline

# Regression shrinkage and selection via the Lasso

*Supervised learning problem:*

- Target variable $y_i$, for cases $i = 1, 2, \ldots n$, features $x_{ij}$, $j = 1, 2, \ldots p$
- The lasso minimizes

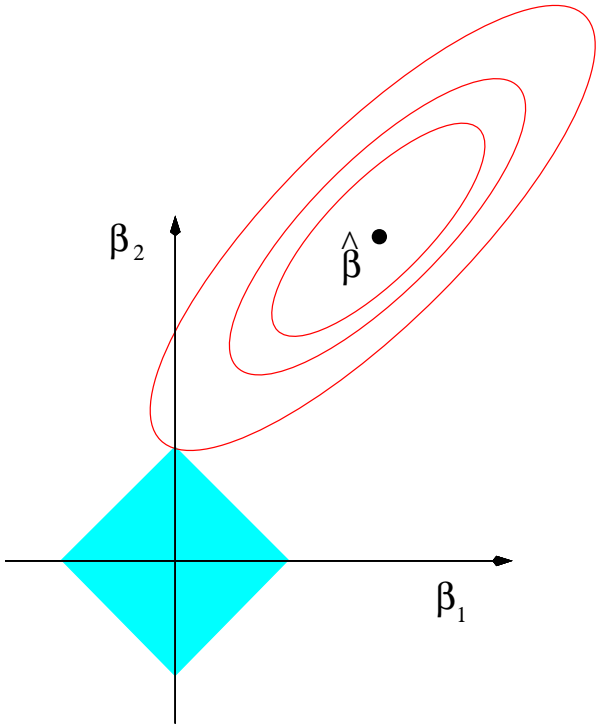
$$\frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

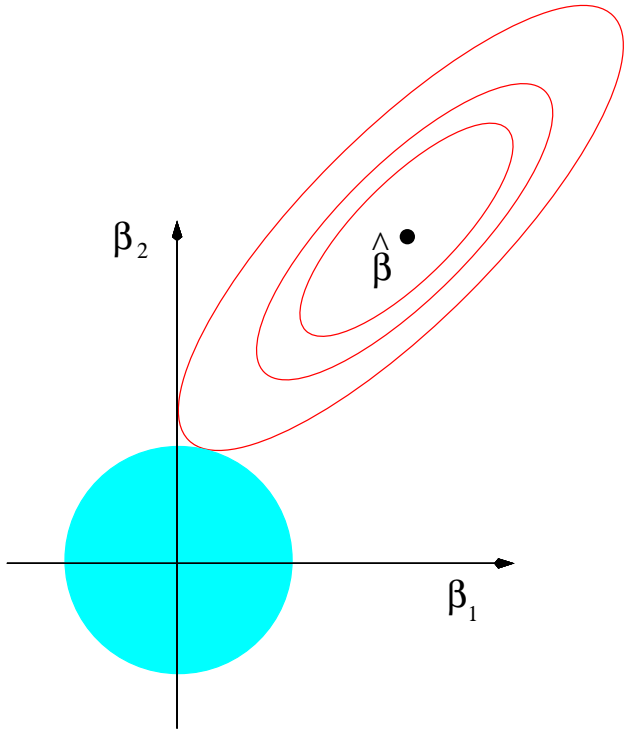
- $\lambda$ is a hyperparameter, chosen by cross-validation. $\lambda = 0$ yields least squares; as $\lambda \to \infty$, the solution $\hat{\beta}$ shrinks to zero.
- Equivalent to minimizing sum of squares with constraint $\sum |\beta_j| \leq s$.

# Lasso shrinkage is special

- Similar to **ridge regression**, which has constraint $\sum_j \beta_j^2 \leq t$
- Lasso does variable selection and shrinkage; ridge regression, in contrast only shrinks.
- Lasso uses an $\ell_1$ norm for the penalty: the $\ell_0$ norm corresponds to best subset selection. The $\ell_1$ norm is convex—a big computational advantage.

# Why does the lasso give a sparse solution?

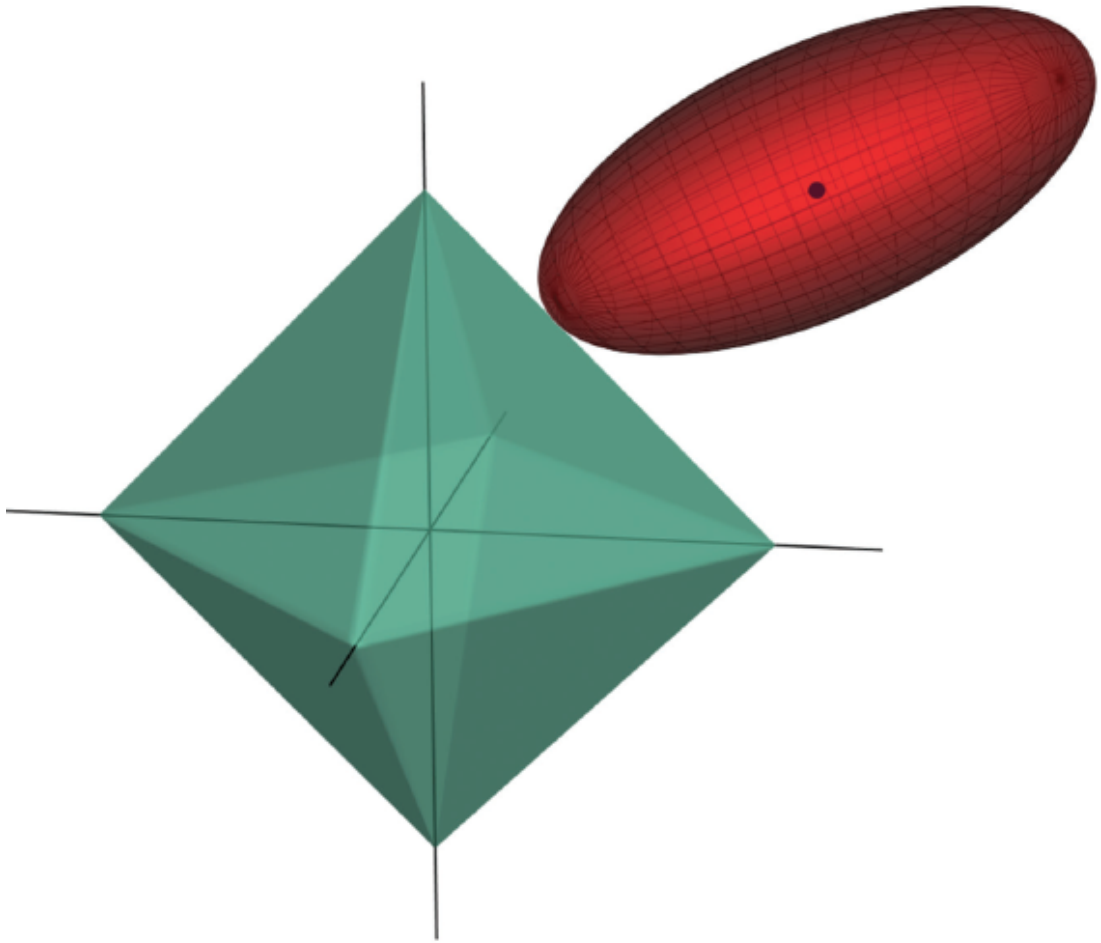



Lasso $\sum_j |\beta_j| \leq s$

Ridge $\sum_j \beta_j^2 \leq s$

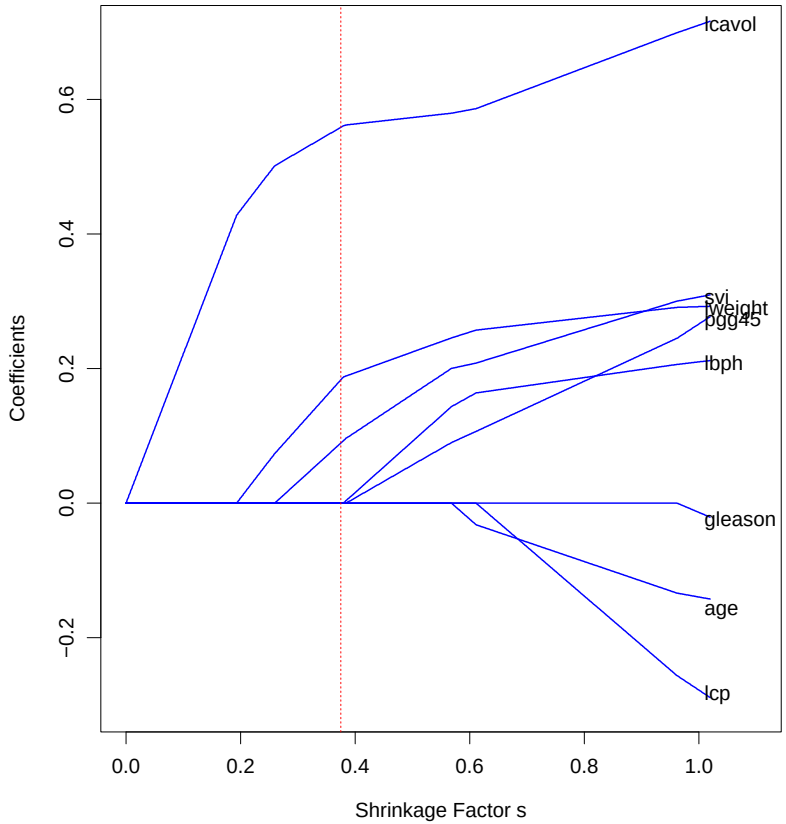
# In 3D



From book "Sparsity in Statistics: the Lasso and its Generalizations" by Hastie, Tibshirani & Wainwright;

# Example: Prostate Cancer Data

$y_i = \log$ (PSA), $x_{ij}$ measurements on a man and his prostate




We estimate the best value of the path parameter $\lambda$ (equivalent to the shrinkage factor in the figure) using a validation set or cross-validation.

# Elastic net (Zou and Hastie)

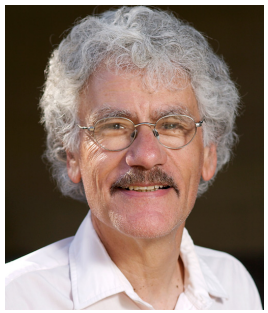
- Minimize

$$\frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda[\alpha\sum_{j=1}^{p}|\beta_j| + (1-\alpha)\cdot\sum_{j=1}^{p}\beta_j^2]$$

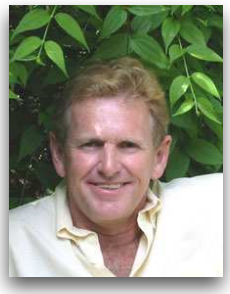
- Can stabilize the model by reducing the effects of correlated features

# The glmnet package in R

- Our lab has written a open-source R language package called `glmnet` for fitting lasso models. Numerics in FORTRAN(!)
- Many clever computational tricks were used to achieve it's impressive speed.
- 4million downloads as of July 2022
- good software also available in Python (e.g. scikit.learn), JMP and SAS



Jerry Friedman



Trevor Hastie

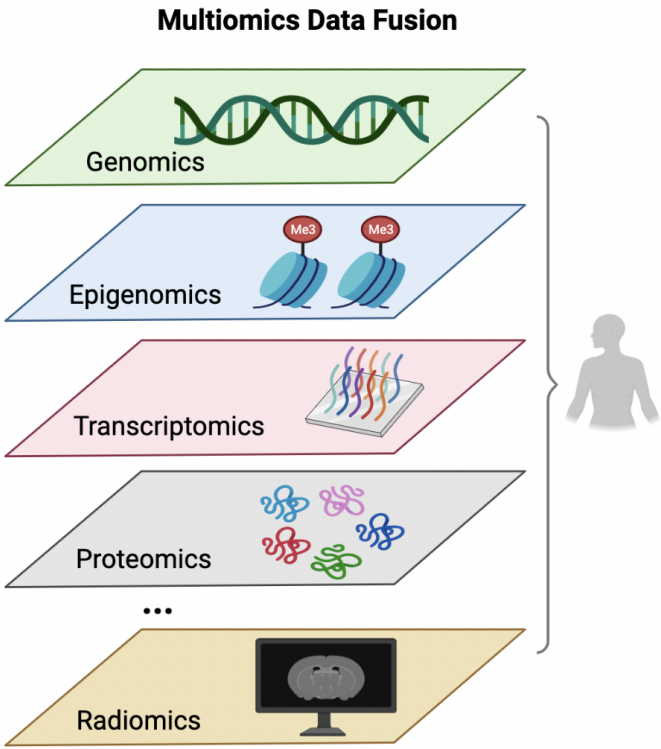
Balasubramanian Narasimhan

# Features of the current version (glmnet 4.1)

- Gaussian, binomial, multinomial, poisson and user-defined "family" objects
- grouped lasso for multi-response Gaussian family
- support for sparse matrices
- feature filtering within cross-validation

# Outline

# Multi-view analysis with "-omics" data is an increasingly important challenge in biology and medicine




**Multi-view analysis:**

The goal is to utilize different feature sets on the same set of observations to model an outcome of interest.

**Why multi-view analysis?**

- Opportunity to gain a more holistic understanding of the outcome of interest;
- Potential for making discoveries that are hidden in a single modality, and achieving more accurate predictions of the outcome.

# Potential application to Covid-19 nowcasting

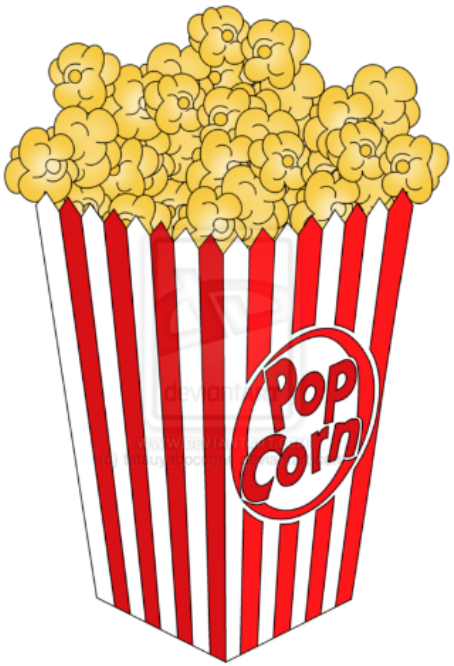
Different views could include

- symptom surveys
- doctors visits
- previous weekly case counts
- health insurance claims $\cdots$

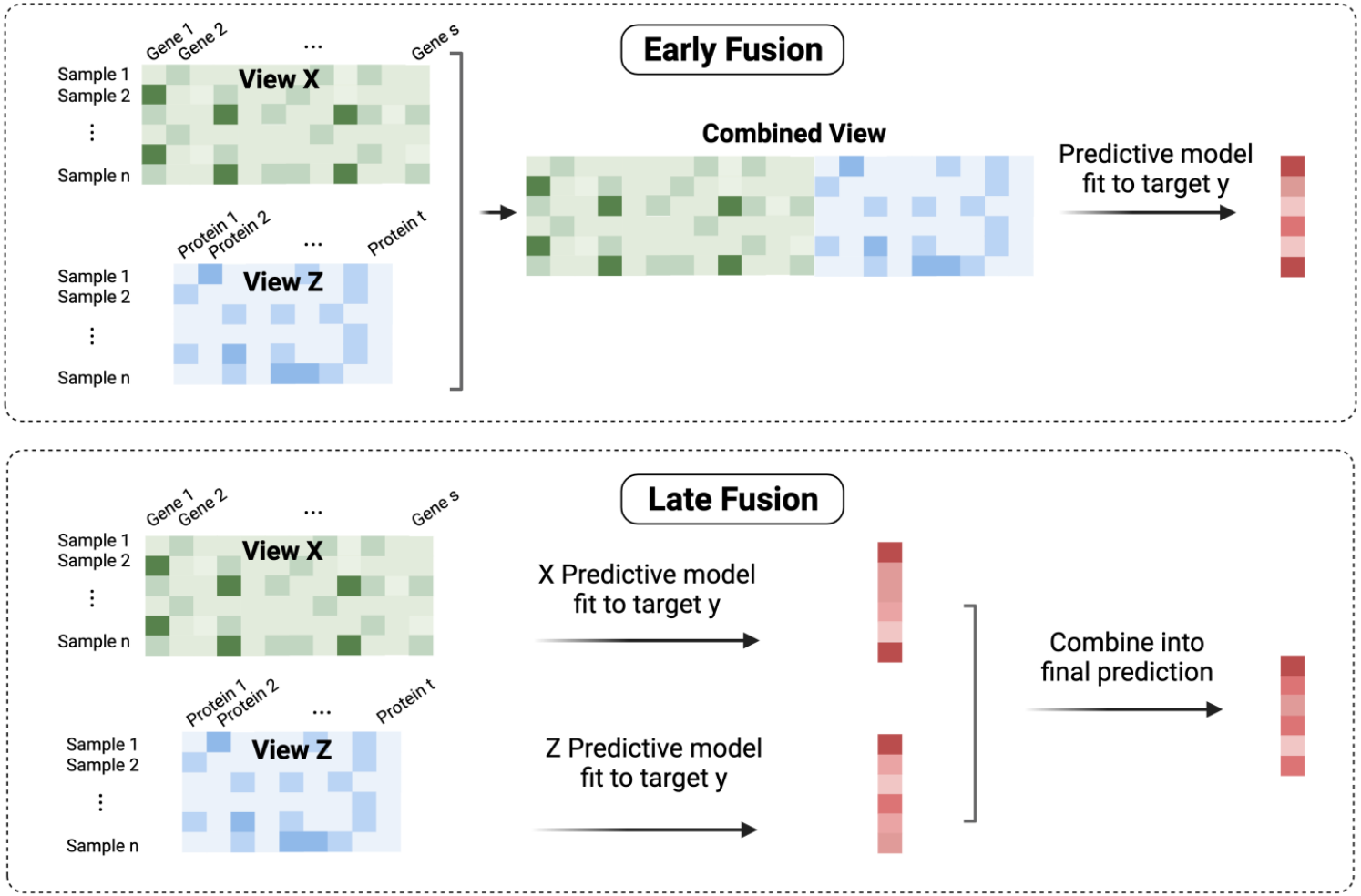
# Cooperative learning for Multiview Analysis

Daisy Ding, Balasubramanian Narasimhan, Shuangning Li, Rob Tibshirani; PNAS September 2022

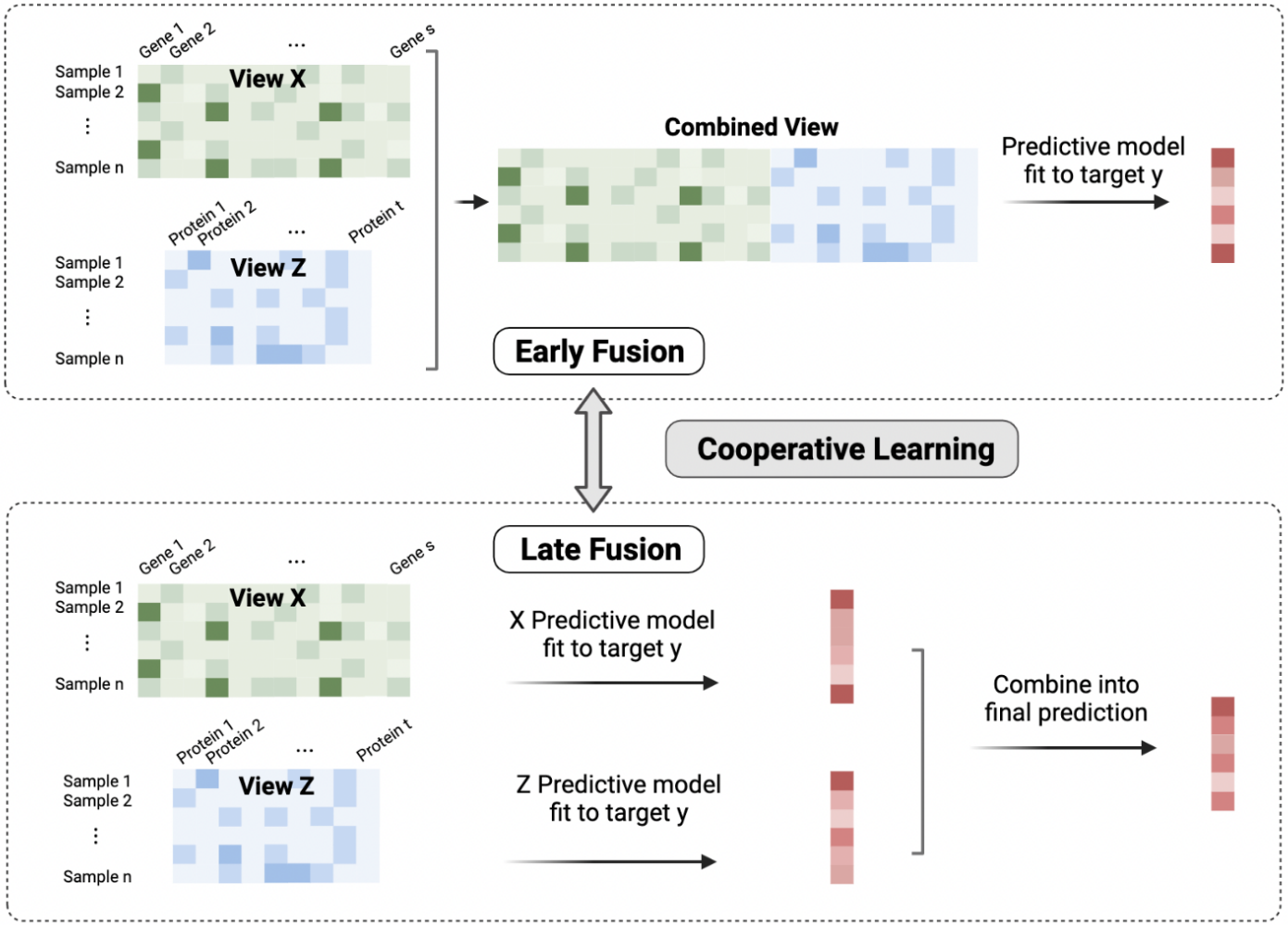
The idea in 3 minutes .



# Existing approaches: early and late fusion



# We propose a new method called cooperative learning that encompasses the early and late fusion



# Outline

# Cooperative learning

For simplicity, will explain everything for the case of just two views; extensions to more than two views are easy

# Cooperative learning

Let $X \in \mathcal{R}^{n \times p_x}$, $Z \in \mathcal{R}^{n \times p_z}$— representing two data views — and $\boldsymbol{y} \in \mathcal{R}^n$ be a real-valued random variable (the target). Fixing the hyperparameter $\rho \geq 0$, we propose to minimize the population quantity:

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\Big], \tag{1}$$

where $f$ is some class of functions, e.g. a linear function, nonlinear function , or a neural network.

- The first term is the usual prediction error: we use $X$ and $Z$ jointly to model $\boldsymbol{y}$.
- The second term is an "agreement" penalty, encouraging the predictions from different views to agree. This is related to "contrastive learning".

- $\rho$ is a hyperparameter that controls the relative importance of the agreement penalty, and we choose it by cross-validation.
- We will see that 0 and 1 are special values. $\rho$ is also not restricted to $[0, 1]$, and in our experiments $\rho > 1$ can sometimes be helpful.

# Two algorithms for cooperative learning

Fixing $\rho \geq 0$, we minimize:

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\Big], \qquad (2)$$

where $f$ is some class of function, e.g. a linear function or a neural network.

## 1. Direct Algorithm

Specialized for cooperative regularized linear regression

## 2. One-at-a-time Algorithm

Modular, allowing one to customize the learner for each view

# Cooperative regularized linear regression

In the setting of regularized regression, cooperative learning combines the lasso penalty with the agreement penalty:

$$J(\theta_x, \theta_z) = \frac{1}{2}||y - X\theta_x - Z\theta_z||^2 + \frac{\rho}{2}||(X\theta_x - Z\theta_z)||^2 + \lambda \cdot (||\theta_x||_1 + ||\theta_z||_1). \quad (3)$$

**When will cooperative learning be helpful?**

Our proposal can be especially powerful when the different data views **share some underlying relationship** that can be leveraged to strengthen signal, while each data view also has its **idiosyncratic noise** that needs to be reduced.

# Simple solution

- This problem can be solved by forming augmented matrices. $X^*, y^*$ and running glmnet on them.
- When $\rho = 0$ we get early fusion; surprisingly, $\rho = 1$ gives a simple form of late fusion where we fit separate models to $X$ and $Z$ and average the predictions.

# Outline

Table: Multiomics studies on labor onset prediction.

| **Methods** | **Test MSE** | | **Ave Number of Features Selected** |
|---|---|---|---|
| | Mean | Std | |
| Separate Proteomics | 475.51 | 80.89 | 26 |
| Separate Metabolomics | 381.13 | 36.88 | 11 |
| Early fusion | 406.37 | 44.77 | 15 |
| Late fusion | 493.34 | 63.44 | 21 |
| **Cooperative learning** | **335.84** | **35.51** | 52 |

Hyperparameter $\rho$ was selected to be 0.5

# New Multiview R Package

## Basic fit

```
library(multiview)
fit <- multiview(list(x,z), y, family=gaussian(), rho=0.5)
coef_ordered(fit, s=0.1)
```

```
##      view view_col standardized_coef        coef
## 13  View1       13        16.5586209  16.5586209
## 25  View1       25       -15.4005249 -15.4005249
## 5   View1        5        12.7521081  12.7521081
```

## Cross-validation

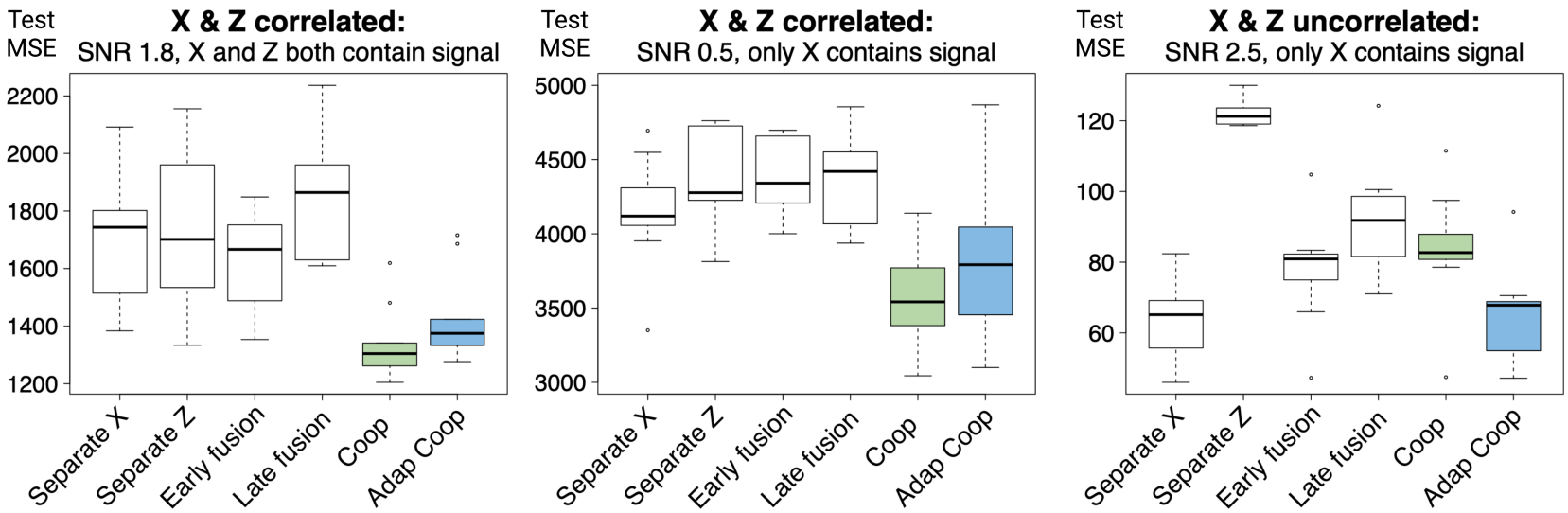
```
cvfit <- cv.multiview(list(x,z), y, family=gaussian(),
                      rho=0.1, nfolds=10)
plot(cvfit)
predict(cvfit, newx=list(x[1:5,],z[1:5,]), s="lambda.min")
```

## Evaluating the contribution of data views in making predictions

```
view.contribution(x_list=list(x=x, z=z), y,
                  x_list_test=list(x=test_x, z=test_z),
                  test_y=test_y, family=gaussian(),
                  rho = 0.5, eval_data="test")
```

```
##                 view    error percentage_improvement
## 1               null 67.13278                0.00000
## 2                  x 46.17828               31.21352
## 3                  z 55.06684               17.97325
## 4 cooperative (all) 44.33946               33.95259
```

# An illustrative simulation study




When the data views are correlated, cooperative learning offers significant performance gains over the early and late fusion methods, by encouraging the predictions from different views to agree.

# Cooperative learning: one-at-a-time algorithm

The solution to

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\Big]$$

has fixed points:

$$\begin{aligned} f_X(X) &= \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_Z(Z)}{(1+\rho)}|X\Big], \\ f_Z(Z) &= \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_X(X)}{(1+\rho)}|Z\Big]. \end{aligned} \tag{4}$$

**Remark 1: Modularity**

The fitting procedure is modular, so that we can choose a fitting mechanism appropriate for each data view:

- For *quantitative features* like gene expression or methylation: regularized regression (lasso, elastic net), boosting, or random forests.
- For *images*: a convolutional neural network.
- For *time series data*: an auto-regressive model or a recurrent neural network.

# What if we have data views from distinct modalities?

Here we have two data views of more distinct data modalities, such as imaging and omics data. Assume our task is to use the imaging and omics data to predict if a patient has a certain disease.

We tailor the fitter suitable to each view, i.e. convolutional neural networks (CNN) for images and lasso for omics. Here we use the one-at-a-time modular algorithm.

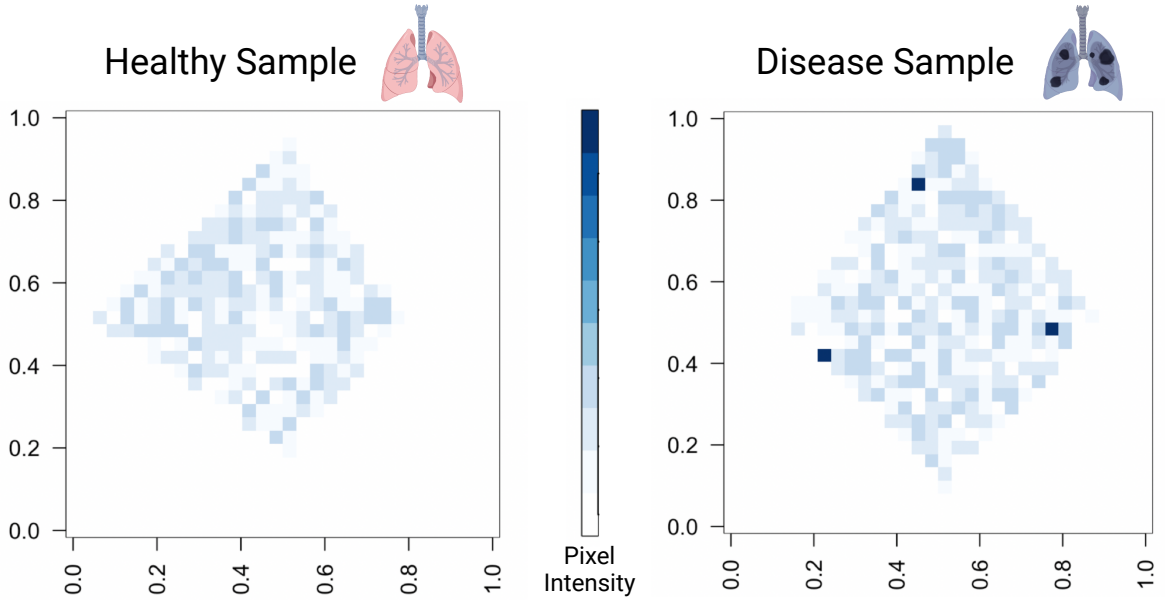
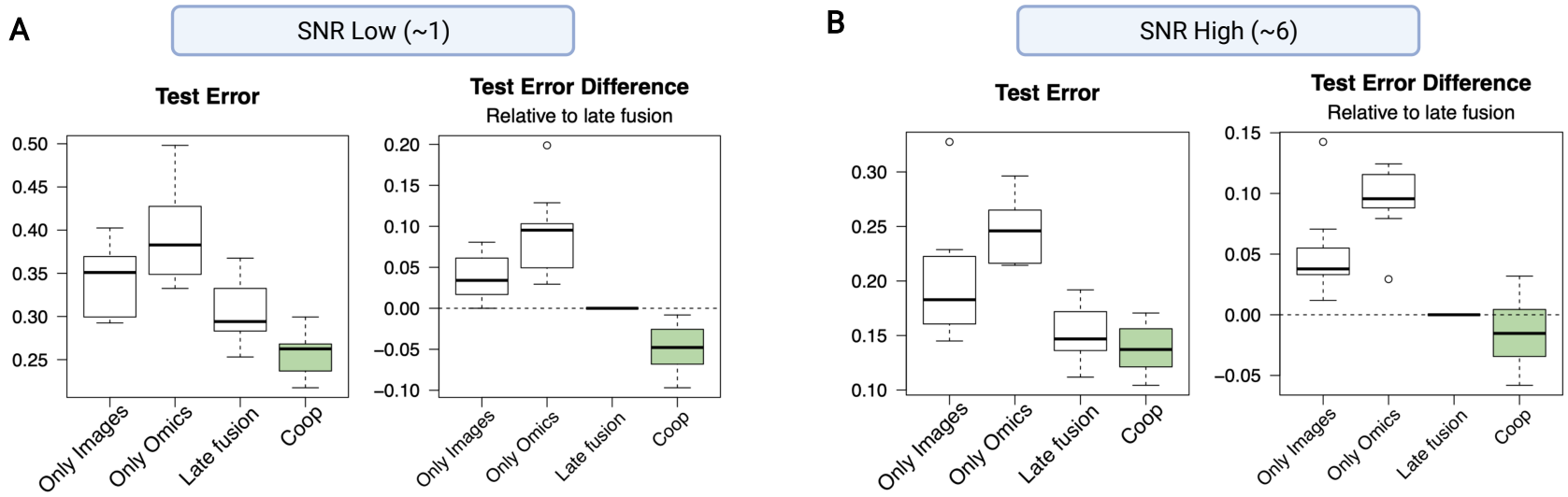



*Figure: Generated images for "healthy" and "disease" samples.*

# Results of simulation study with imaging and "omics" data




1. Late fusion achieves a lower test error than the separate models;
2. Cooperative learning outperforms late fusion and achieves the lowest test error by encouraging the predictions from the two views to agree;
3. Cooperative learning is especially helpful when SNR is low.

# Outline

# Extensions and takeaways

- Can make use of some unlabelled samples
- Can handle missing data naturally
- Can make exploit paired features
- R package `multiview` on CRAN