High-dimensional data analysis

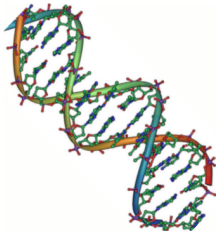
Nicolai Meinshausen
Seminar für Statistik, ETH Zürich

Van Dantzig Seminar, Delft
31 January 2014

Historical start: Microarray data (Golub et al., 1999)

Gene expression levels of more than 3000 genes are measured for n = 72 patients, either suffering from acute lymphoblastic leukemia ("X", 47 cases) or acute myeloid leukemia ("O", 25 cases). Obtained from Affymetrix oligonucleotide microarrays.

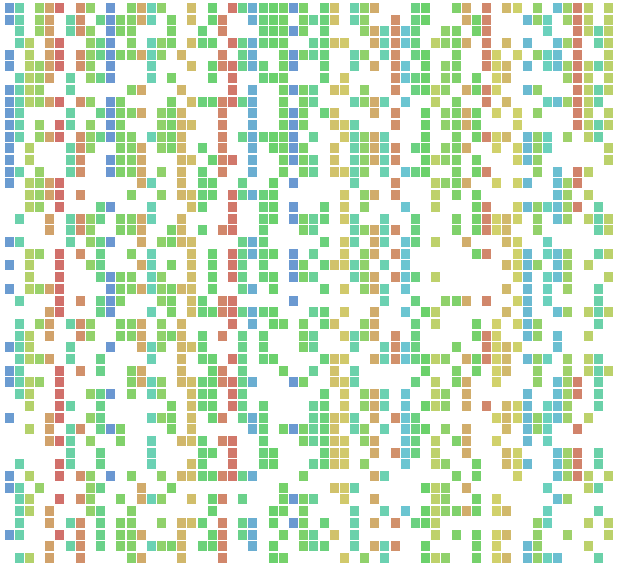
Gene expression analysis

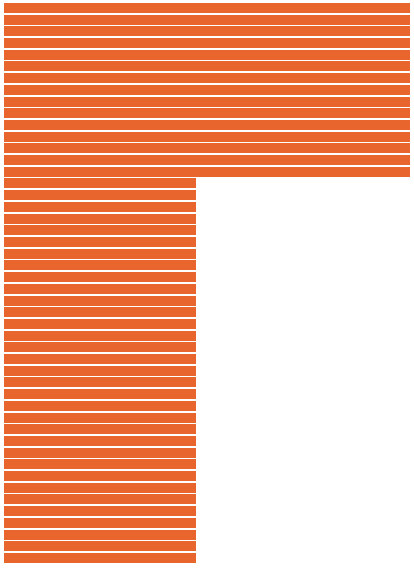


1000-20000
genes

cancer
(sub-)type

100-1000
people





Large-scale inference problems

	sample size	predictor variables	goal
gene expression	hundreds of people	thousands of genes	predict cancer (sub-)type
webpage ads	millions to billions of webpages	billions of word- and word-pair frequencies	predict click-through rate
credict card fraud	thousands to billions of transactions	thousands to billions information pieces about transaction/customer	detect fraudulent transactions
medical data	thousands of people	tens of thousands to billions of indicators for symptoms/drug-use	estimate risk of stroke
particle physics	millions of particle collisions	millions of intensity measurements	classify type of particles created

Inference "works" if we need just a small fraction of variables to make a prediction (but do not yet know which ones).

High-dimensional data

Let Y be a real-valued response in $\mathbb{R}^n$ (binary for classification), X a $n \times p$-dimensional design and assume a linear model in which

$$Y = X\beta^* + \varepsilon + \delta,$$
$$P(Y = 1) = f(X\beta^* + \delta), \qquad \text{where } f(x) = 1/(1 + \exp(-x))$$

for some (sparse) vector $\beta^* \in \mathbb{R}^p$, noise $\varepsilon \in \mathbb{R}^n$ and model error $\delta \in \mathbb{R}^n$. Regression (or classification) is high-dimensional if $p \gg n$.

Basis Pursuit (Chen et al. 99) and Lasso (Tibshirani 96)

Let Y be the n-dimensional response vector and X the $n \times p$-dimensional design.
Basis Pursuit (Chen et al., 99)

$$\hat{\beta} = \text{argmin } \|\beta\|_1 \text{ such that } Y = X\beta.$$

Lasso:

$$\hat{\beta}^{\tau} = \text{argmin } \|\beta\|_1 \text{ such that } \|Y - X\beta\|_2 \leq \tau.$$

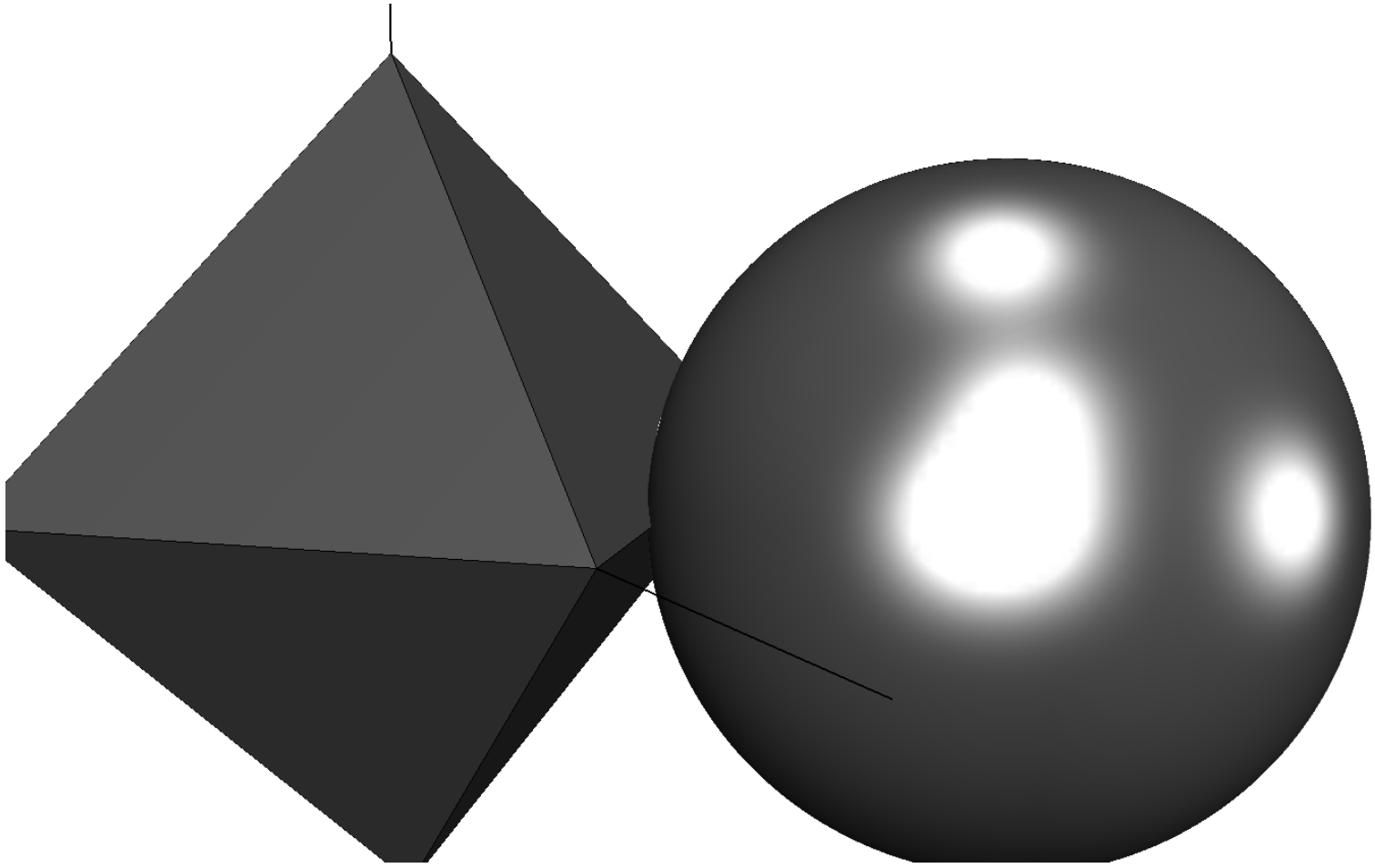
Equivalent to (Tibshirani, 96):

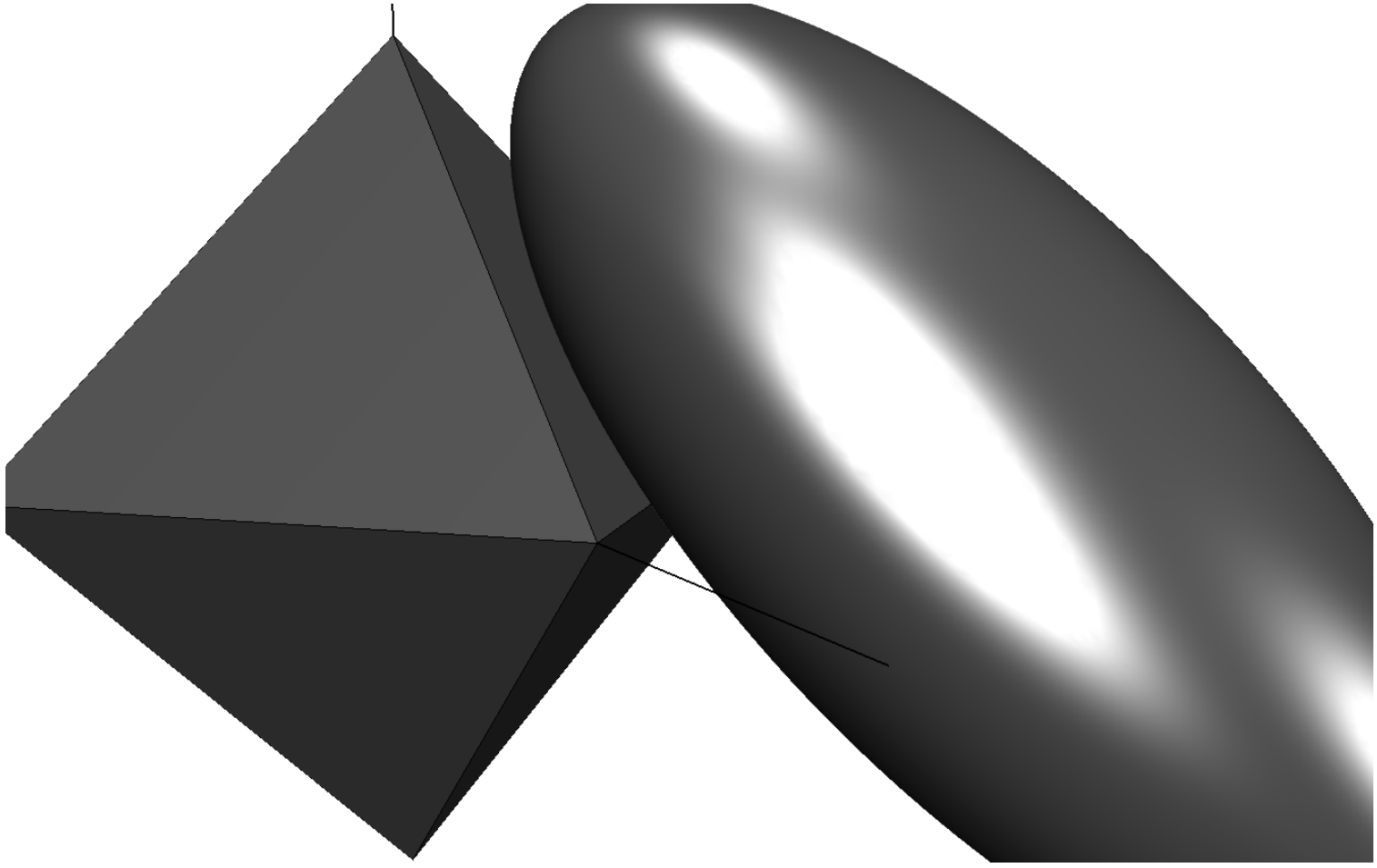
$$\hat{\beta}^{\lambda} = \text{argmin } \|Y - X\beta\|_2 + \lambda\|\beta\|_1.$$

Combines sparsity (some $\hat{\beta}$-components are 0) and convexity. Many variations exist.

Two important properties:

- Mixing two equally good solutions always improves the fit (as loss function is convex)
- Mixing solutions produces another valid solution (as feasible sets are convex)





When does it work?

- For *prediction* oracle inequalities in the sense that

$$\|X(\hat{\beta} - \beta^*)\|_2^2/n \leq c \log(p) \frac{\sigma^2 s}{n}$$

for some constant $c > 0$, need *Restricted Isometry Property* (Candes, 2006) or weaker *compatibility condition* (Geer, 2008). Slower convergence rates possible with weaker assumptions (Greenstein and Ritov, 2004).

When does it work?

- For *prediction* oracle inequalities in the sense that

$$\|X(\hat{\beta} - \beta^*)\|_2^2/n \leq c \log(p) \frac{\sigma^2 s}{n}$$

  for some constant $c > 0$, need *Restricted Isometry Property* (Candes, 2006) or weaker *compatibility condition* (Geer, 2008). Slower convergence rates possible with weaker assumptions (Greenstein and Ritov, 2004).

- For correct variable selection in the sense that

$$P\Big(\exists \lambda : \{k : \hat{\beta}_k^\lambda \neq 0\} = \{k : \beta_k^* \neq 0\}\Big) \approx 1,$$

  need strong *irrepresentable* (Zhao and Yu, 2006) or *neighbourhood stability* condition (NM and Bühlmann, 2006).

Compatibility condition

The usual minimal eigenvalue of the design

$$\min\{\|X\beta\|_2^2 : \|\beta\|_2 = 1\}$$

always vanishes for high-dimensional data with $p > n$.

Compatibility condition

The usual minimal eigenvalue of the design

$$\min\{\|X\beta\|_2^2 : \|\beta\|_2 = 1\}$$

always vanishes for high-dimensional data with $p > n$.

The ϕ be the (L, S)-restricted eigenvalue (Geer, 2007):

$$\phi^2(L, S) = \min\{s\|X\beta\|_2^2 : \|\beta_S\|_1 = 1 \text{ and } \|\beta_{S^c}\|_1 \leq L\},$$

where $s = |S|$ and $(\beta_S)_k = \beta_k 1\{k \in S\}$.

1. If $\phi(L, S) > c > 0$ for some $L > 1$, then we get oracle rates for prediction and convergence of $\|\beta^* - \hat{\beta}^\lambda\|_1$.
2. If $\phi(1, S) > 0$ and $f = X\beta^*$ for some β^* with $\|\beta^*\|_0 \leq s$, then the following two are identical

$$\text{argmin}\|\beta\|_0 \text{ such that } X\beta = f$$
$$\text{argmin}\|\beta\|_1 \text{ such that } X\beta = f.$$

1. If $\phi(L, S) > c > 0$ for some $L > 1$, then we get oracle rates for prediction and convergence of $\|\beta^* - \hat{\beta}^\lambda\|_1$.
2. If $\phi(1, S) > 0$ and $f = X\beta^*$ for some β^* with $\|\beta^*\|_0 \leq s$, then the following two are identical

$$\text{argmin}\|\beta\|_0 \text{ such that } X\beta = f$$
$$\text{argmin}\|\beta\|_1 \text{ such that } X\beta = f.$$

The latter equivalence requires otherwise the stronger *Restricted Isometry Property* which implies that $\exists \delta < 1$ such that

$$\forall b \text{ with } \|b\|_0 \leq s: \qquad (1-\delta)\|b\|_2^2 \leq \|Xb\|_2^2 \leq (1+\delta)\|b\|_2^2,$$

which can be a useful assumption for random designs X, as in compressed sensing.

Three examples:

1. Compressed sensing
2. Electro-retinography
3. Mind reading

Compressed sensing

High quality JPEG
File Size: 77.9 kb

Medium quality JPEG
File Size: 19.11 kb

Images are often sparse after taking a wavelet transformation X:

$$u = Xw, \qquad \text{where}$$

- $w \in \mathbb{R}^n$: original image as n-dimensional vector
- $X \in \mathbb{R}^{n \times n}$: wavelet transformation
- $u \in \mathbb{R}^n$: vector with wavelet coefficients

High quality JPEG
File Size: 77.9 kb

Medium quality JPEG
File Size: 19.11 kb

Original wavelet transformation:

$$u = Xw, \qquad \text{where}$$

The wavelet coefficients u are often sparse in the sense that it has only a few large entries. Keeping just a few of them allows a very good reconstruction of the original image w.
Let $\tilde{u} = u1\{|U| \geq \tau\}$ be the hard-thresholded coefficients (easy to store).
Then re-construct image as $\tilde{w} = X^{-1}\tilde{u}$.

Conventional way:

- measure image w with 16 million pixels
- convert to wavelet coefficients $u = Xw$
- throw away most of u by keeping just the largest coefficients

Is efficient as long as pixels are cheap.

For situations where pixels are expensive (different wavelengths, MRI) can do compressed sensing: observe only

$$y = \Phi u = \Phi(Xw),$$

where for $q \ll n$, matrix $\Phi \in \mathbb{R}^{q \times n}$ has iid entries drawn from $\mathcal{N}(0, 1)$. One entry of q-dimensional vector y is thus observed by a random transformation of the original image.

Each random mask corresponds to one row of Φ.

Reconstruct u by Basis Pursuit:

$$\hat{u} = \text{argmin} \|\tilde{u}\|_1 \text{ such that } \Phi\tilde{u} = y.$$

Observe

$$y = \Phi u = \Phi(Xw),$$

where for $q \ll n$, matrix $\Phi \in \mathbb{R}^{q \times n}$ has iid entries drawn from $\mathcal{N}(0, 1)$. Reconstruct wavelet coefficients u by Basis Pursuit:

$$\hat{u} = \text{argmin} \|\tilde{u}\|_1 \text{ such that } \Phi\tilde{u} = y.$$

Observe

$$y = \Phi u = \Phi(Xw),$$

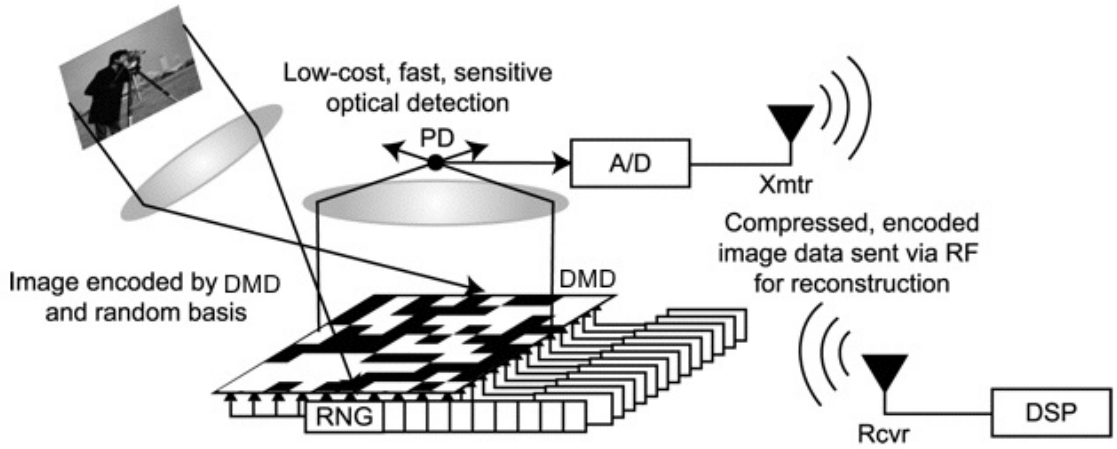
where for $q \ll n$, matrix $\Phi \in \mathbb{R}^{q \times n}$ has iid entries drawn from $\mathcal{N}(0, 1)$. Reconstruct wavelet coefficients u by Basis Pursuit:

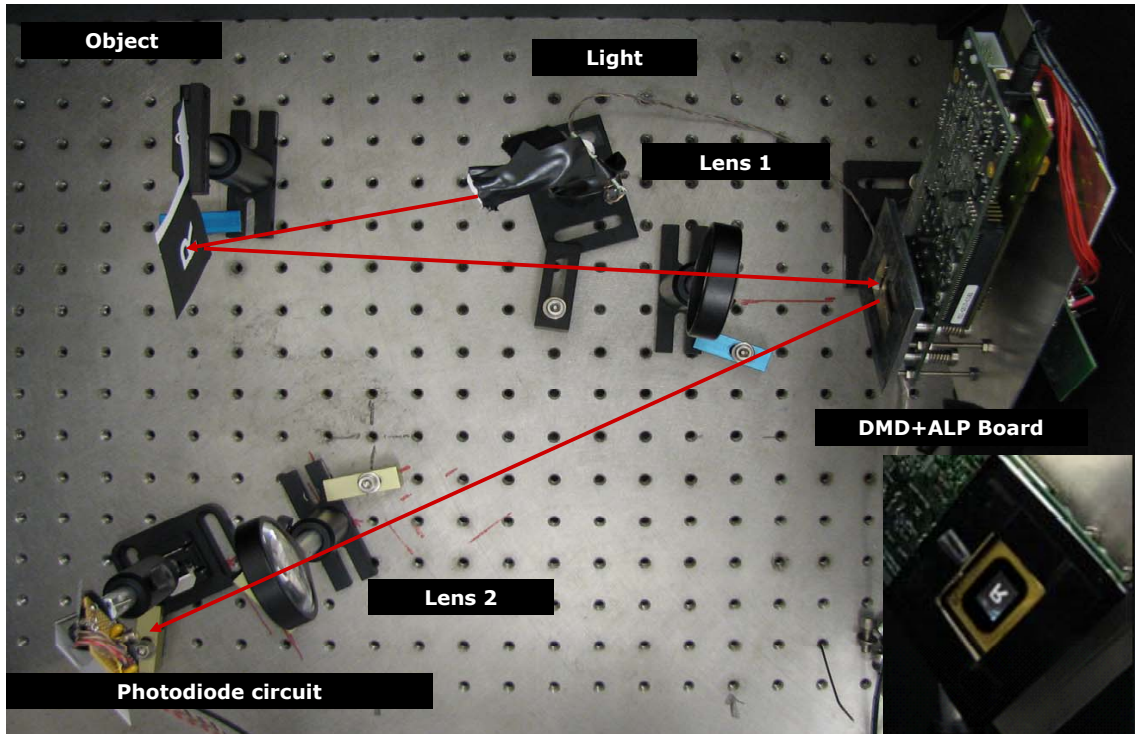
$$\hat{u} = \text{argmin} \|\tilde{u}\|_1 \text{ such that } \Phi\tilde{u} = y.$$

Matrix Φ satisfies for $q \geq s \log(p/s)$ with high probability the *Random Isometry Property*, including the existence of a $\delta < 1$ such that (Candes, 2006) for all s-sparse vectors

$$(1 - \delta)\|b\|_2^2 \leq \|\Phi b\|_2^2 \leq (1 + \delta)\|b\|_2^2.$$

Hence, if original wavelet coeffcients are s-sparse, we only need to make of order $s \log(n/s)$ measurements to recover u exactly (with high probability)!



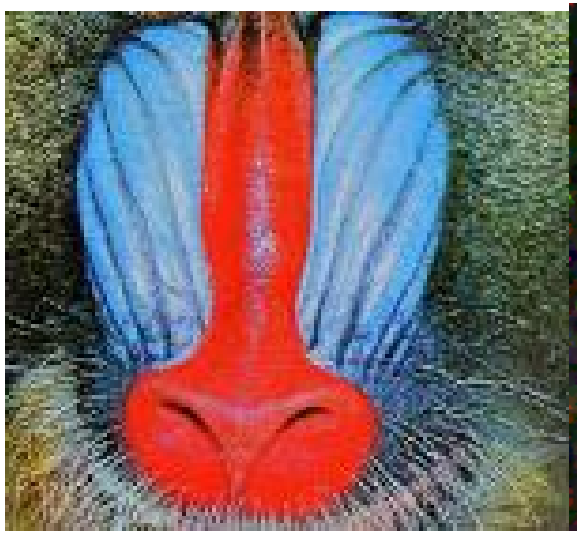


dsp.rice.edu/cs/camera

Original

16384 Pixels
1600 Measurements
(10%)

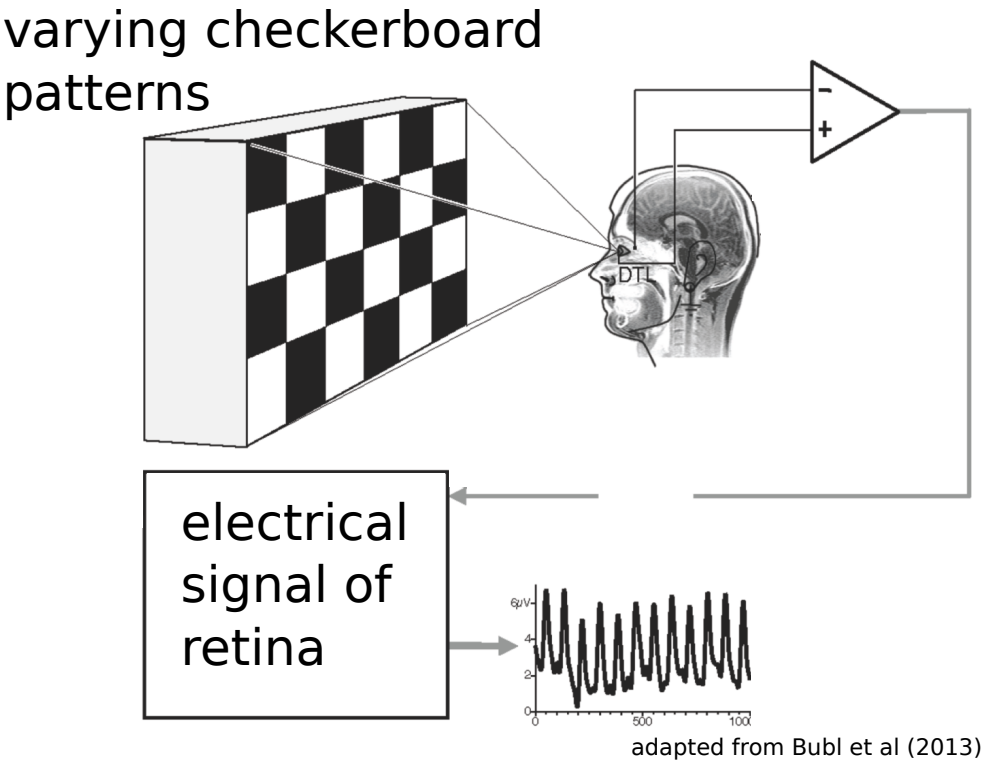
16384 Pixels
3300 Measurements
(20%)



dsp.rice.edu/cs/camera

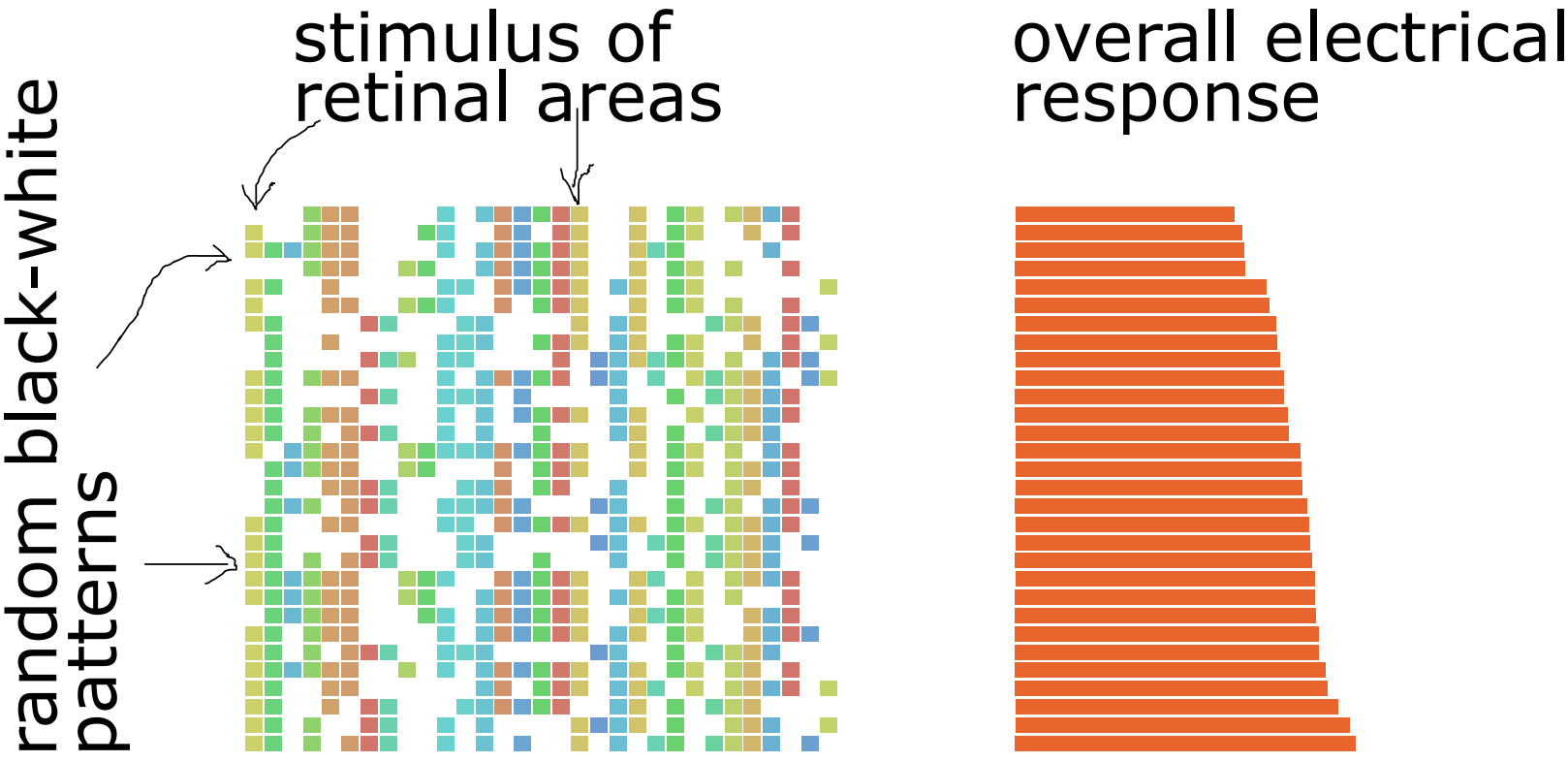
Retina Checks (Electroretinography)

Can one identify “blind” spots on the retina while measuring only the aggregate electrical signal ?



adapted from Bubl et al (2013)

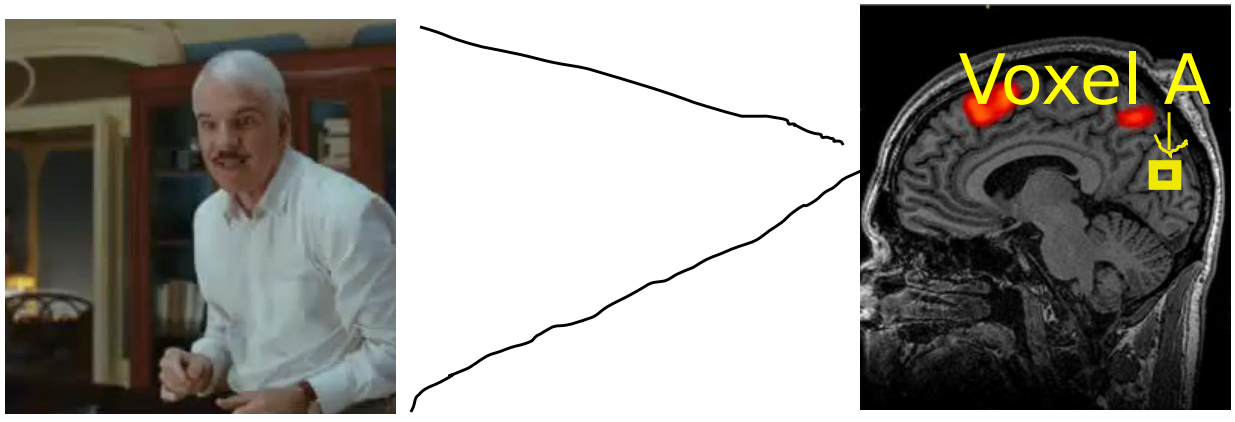
Assume there are p retinal areas (corresponding to the blocks in the shown patterns) of which some can be unresponsive.



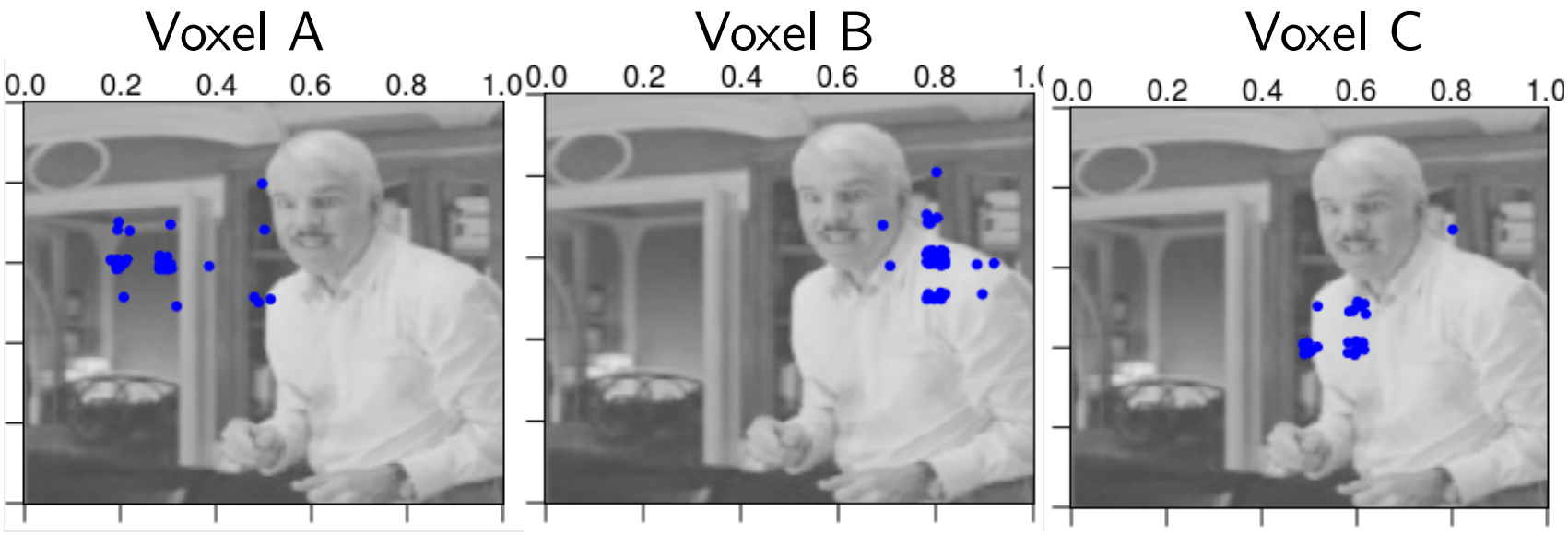
Can detect s unresponsive retinal areas with just $s \log(p/s)$ random patterns.

Mind reading

Can use Lasso-type inference to infer for a single voxel in the early visual cortex which stimuli lead to neuronal activity using fmri-measurements (Nishimoto et al., 2011 at Gallant Lab, UC Berkeley).

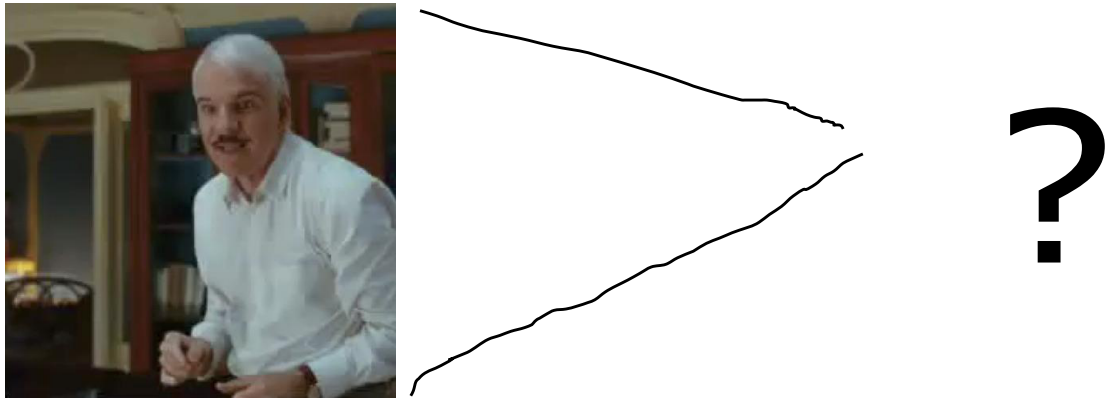


Show movies and detect which parts of the image a particular voxel of 100k neurons is sensitive to.

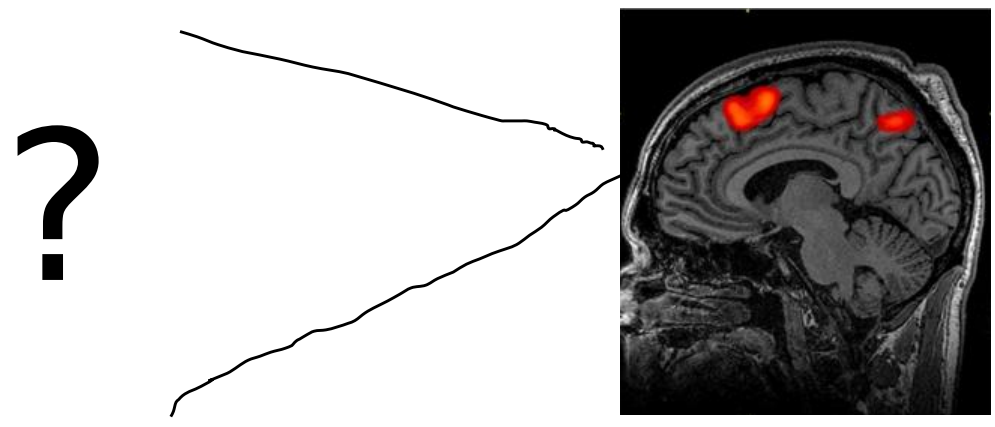


Learn a Lasso regression that predicts neuronal activity in each separate voxel. Dots indicate large regression coefficients and thus important regions for a voxel.

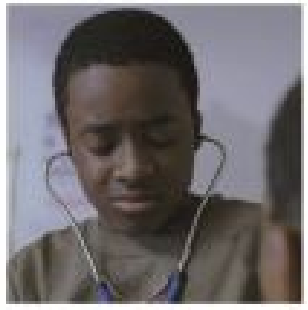
Allows to forecast brain activity at all voxels, given an image.

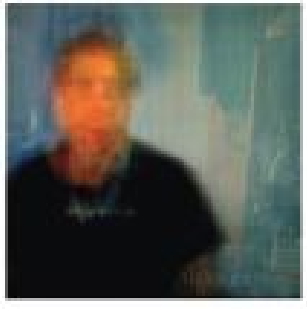


Given only brain activity, can reverse the process and ask which image best explains the neuronal activity (given the learned regressions).






Four challenges:

- Trade-off between statistical and computational efficiency
- Inhomogeneous data
- Confidence statements
- Interactions in high dimensions

Interactions

Many datasets are only moderately high-dimensional with raw data

- Activity of approximately 20k genes in microarray data
- Presence of about 20k words in texts/websites
- About 15k different symptoms and 15k different drugs recorded in medical histories (US).

Interactions look for effects that are caused by simultaneous presence of two or more variables.

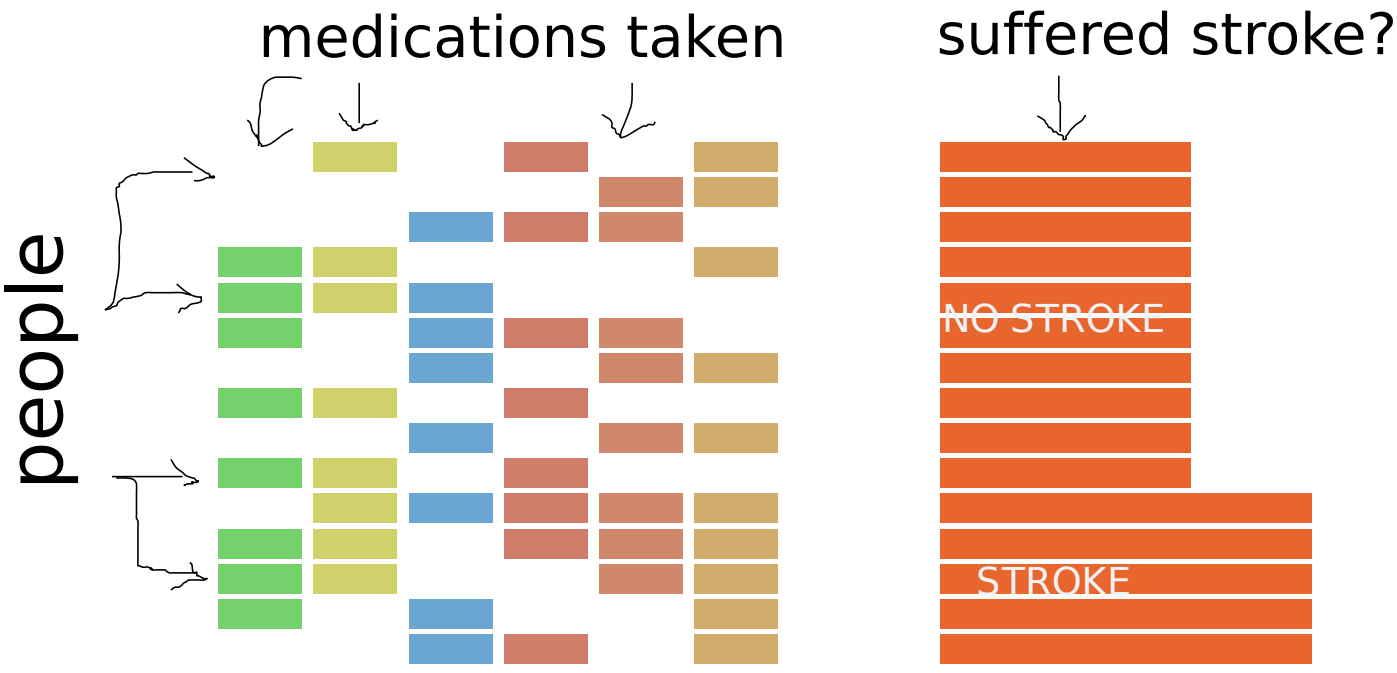
- are two or more genes active at the same time ?
- do two words appear close together ?
- have two drugs been taken simultaneously ?

Medical data

OMOP: Observational Medical Outcomes Project (omop.org)

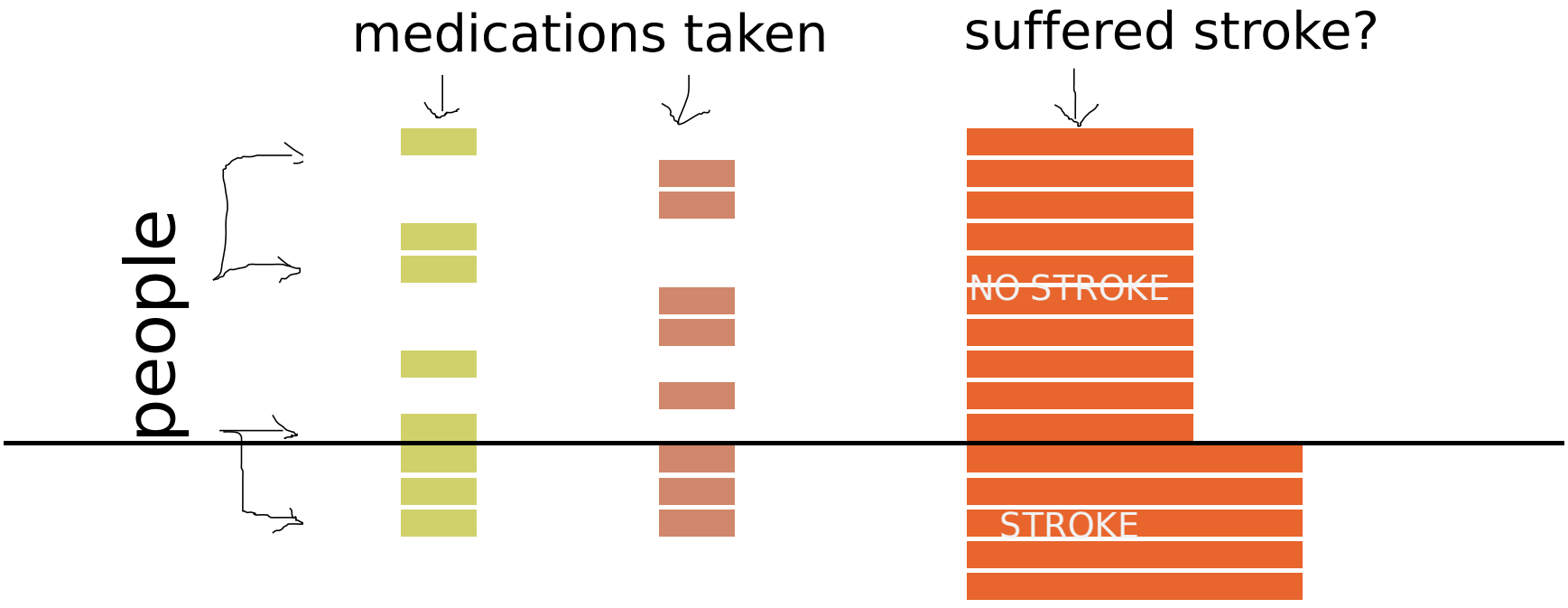
1. Collect medical information (drugs taken, symptoms diagnosed) for 100.000 patients
2. In total, about 15.000 drugs and 15.000 distinct symptoms encoded.

Try to detect drug-drug interactions or make risk assesments based on medical data:
Is drug A changing the risk of a stroke if taken together with drug B ?



Toy data for 10 “patients” (instead of 10k) with six drugs (instead of 15k).
Is there a pattern that differentiates the stroke from the non-stroke patients?

Try to detect drug-drug interactions or make risk assesments based on medical data:
Is drug A changing the risk of a stroke if taken together with drug B ?

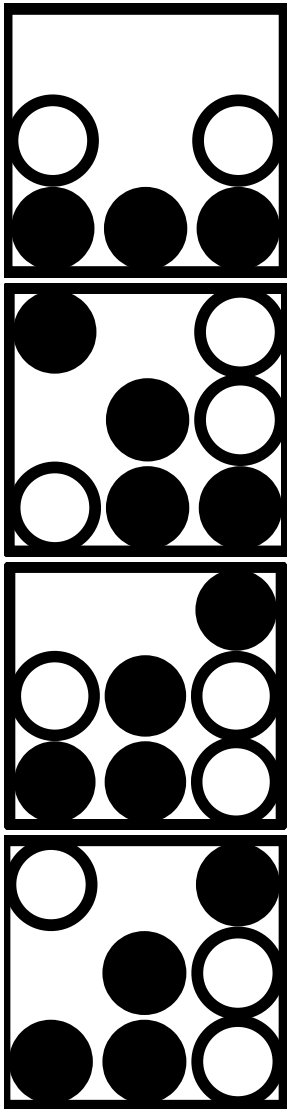


Toy data for 10 “patients” (instead of 10k) with six drugs (instead of 15k).
Is there a pattern that differentiates the stroke from the non-stroke patients?

Can generate very high-dimensional data quickly if expanding interactions as new dummy variables.
Cannot check all interactions as there are already $> 10^{12}$ interactions of third order (for $p \approx 30k$). If checking hundred third-order interaction per second, it would take more than 1400 years for a single dataset.

Can beat the complexity of $O(p^s)$ when seaching for interactions of order s in certain circumstances.
If data are sufficiently sparse, we can search over observations, not variables (*Random Intersection Trees*, Shah & NM, 2014), getting a lower computational complexity than with naive search.

Example: Tic-Tac-Toe Data



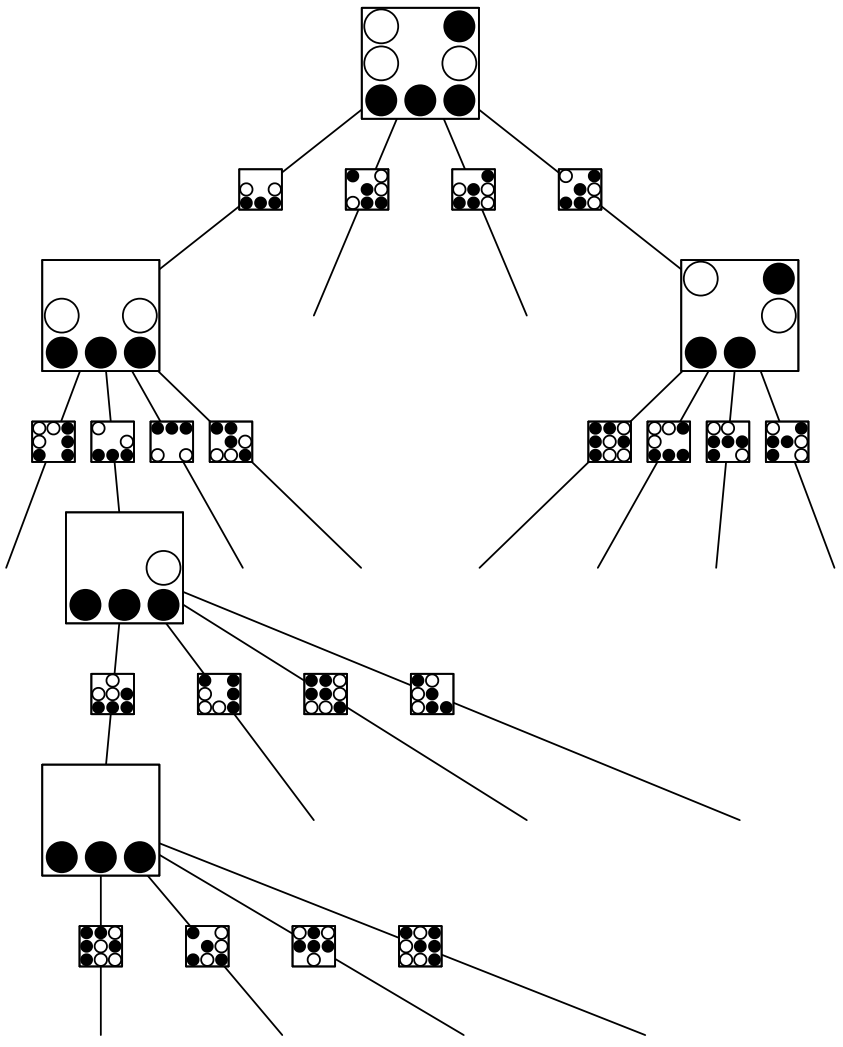
Dataset with endgames of Tic-Tac-Toe games. Learn the rules of the game (or probabilities of winning) by looking at the outcomes of previous games.

Each variables is coded as binary (e.g. "is the first square occupied by a black stone?")

Basic Idea of *Random Intersection Trees*: take a randomly chosen sets of games where black won and look at what the outcomes have in common.

Arranging the search on a tree

Computing intersections is cheap if the sets are already small.



Random Intersection Search Tree.

Intersections are shown in the nodes. Random observations along edges. Stop if pattern becomes too frequent for opposite class (white wins).

Computational complexity depends on the sparsity of the variables and frequency of the interaction but can be as low as $O(p)$ even for $s > 2$.

Four challenges:

- Trade-off between statistical and computational efficiency
- Inhomogeneous data
- Confidence statements
- Interactions in high dimensions

Confidence Intervals for high-dimensional regression

- If prediction is only goal, point estimation of $\beta^* \in \mathbb{R}^p$ is sufficient.
- Often, we want to know exactly which coefficients are really large:
  - which regions really activate a given region in the brain ?
  - which genes are relevant to predict cancer type ?
  - which taken drugs or personal characteristics are influential to predict increased risk of heart attack?

For $p \gg n$, can we get confidence intervals for β^* in

$$Y = X\beta^* + \varepsilon \qquad ?$$

The null-space of X is at least $p - n$-dimensional, and β^* is either the ℓ_0- or ℓ_1-sparsest vector fulfiling $E(Y) = X\beta$.

At least four possible approaches:

- Data-splitting Wasserman and Roeder, 2009; NM, Meier and Buhlmann, 2009
- Residualizing variables Zhang, 2011; Geer, Buhlmann and Ritov, 2013; Javanmard and Montanari, 2013
- Residual-type bootstrap approaches Chatterjee and Lahiri, 2013; Liu and Yu, 2013
- Group-testing NM, 2013

Data-splitting

Wasserman and Roeder, 2009; NM, Meier and Buhlmann, 2009

- Split the data repeatedly into two halfs.
  - Select an initial set $\hat{S} \subset \{1, \ldots, p\}$ of variables on first half
  - Apply classical low-dimensional testing with variables in $\hat{S}$ on second hald of data
- Aggregate p-values by using appropriate quantiles across all splits; for example twice the median (NM, Meier and Buhlmann, 2009).

Appropriate error control if $P(S \subseteq \hat{S})$ large, where $S = \{k : \beta_k^* \neq 0\}$. Generally requires a condition on the minimal non-zero coefficient of β (beta-min condition) and compatibility condition. Quite robust in practice.

Residualizing each variable

Zhang, 2011; Geer et al. 2013; Javanmard and Montanari, 2013

For $p < n$, let

$Z_k =$ residuals of X_k when regressing on all other variables $\{1, \ldots, p\} \setminus k$.

for the OLS solution $\hat{\beta}^{OLS}$,

$$\hat{\beta}_k^{OLS} = \frac{Y^t Z_k}{X_k^t Z_k}$$

Residualizing each variable

Zhang, 2011; Geer et al. 2013; Javanmard and Montanari, 2013

For $p < n$, let

$Z_k =$ residuals of X_k when regressing on all other variables $\{1, \ldots, p\} \setminus k$.

for the OLS solution $\hat{\beta}^{OLS}$,

$$\hat{\beta}_k^{OLS} = \frac{Y^t Z_k}{X_k^t Z_k}$$

Translate to Lasso setting, let Z_k be identical to above, except that the regression is done as a Lasso-regression. Set again

$$\hat{\beta}_k = \frac{Y^t Z_k}{X_k^t Z_k}.$$

Residualizing each variable

Zhang, 2011; Geer et al. 2013; Javanmard and Montanari, 2013

For $p < n$, let

$Z_k =$ residuals of X_k when regressing on all other variables $\{1, \ldots, p\} \setminus k$.

for the OLS solution $\hat{\beta}^{OLS}$,

$$\hat{\beta}_k^{OLS} = \frac{Y^t Z_k}{X_k^t Z_k}$$

Translate to Lasso setting, let Z_k be identical to above, except that the regression is done as a Lasso-regression. Set again

$$\hat{\beta}_k = \frac{Y^t Z_k}{X_k^t Z_k}.$$

Then

$$\hat{\beta}_k = \beta_k^* + \text{known variance} + \text{controllable bias}.$$

Works under the assumption that population covariance Σ of X has minimal eigenvalue, β^* is sparse and Σ^{-1} is sparse.

Two drawbacks of these approaches:

- Assumptions on the design matrix are typically not verifiable.
- Testing of individual variables typically not very fruitful for high correlation between variables.

Group-testing

NM, 2013

Can also get (conservative) confidence intervals for single variables or the effect of whole groups without making an assumption on the design matrix.

Idea: let C be a region for which $P(\varepsilon \in C) = 1 - \alpha$. Then, with probability at least $1 - \alpha$,

$$\beta^* = BP(Y - \varepsilon) \quad \text{for some } \varepsilon \in C$$

where $BP(Y) = \text{argmin}\|\beta\|_1$ such that $X\beta = Y$ is the Basis Pursuit solution.

Find a region C for which $P(\varepsilon \in C) = 1 - \alpha$ is high for a suitable $m = m(n)$ by

$$C = \text{convex hull}(\varepsilon_1, \ldots, \varepsilon_m)$$

and let

$$\hat{\beta}^{(1)}, \ldots, \hat{\beta}^{(m)}$$

be the Basis Pursuit solutions at $Y - \varepsilon_1, \ldots, Y - \varepsilon_m$.
Then $P(\beta^* \in \mathcal{B}) \geq 1 - \alpha$, where

$$\mathcal{B} \in \{\beta : \exists \alpha \in \mathbb{R}^+ \text{such that } X\beta = Y - \sum_{j=1}^{m} \alpha_j \varepsilon_j \text{ and } \|\beta\|_1 \leq \sum_{j=1}^{m} \alpha_j \|\beta^{(j)}\|_1\}$$

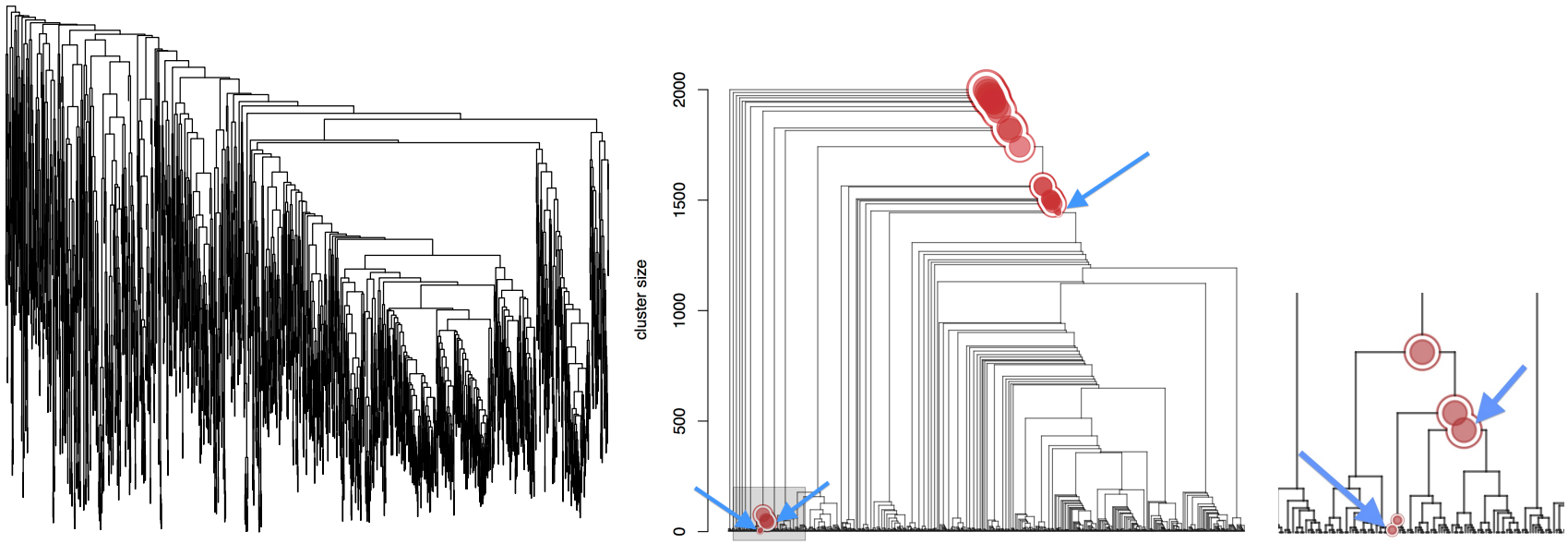
Note that $\mathcal{B}$ is convex.

We have $P(\beta^* \in \mathcal{B}) \geq 1 - \alpha$ for a convex set $\mathcal{B}$.
A lower bound for a group effect $\|\beta^*_G\|_1$ with $G \subseteq \{1, \ldots, p\}$ is then

$$\min_{\beta \in \mathcal{B}} \|\beta_G\|_1,$$

which can be solved by linear programming.
Can also find upper bounds for $\|\beta_G\|_1$ and bounds for $\|\beta^*_G\|_2$ by quadratic programming. Unknown noise can be dealt with by sample splitting.

Example: result for Riboflavin production expression data with $p = 2000$ and $n = 115$.



Most methods implemented in R-package `hdi` on R-forge (hopefully soon CRAN).