

Fondements de l'Intelligence Artificielle et du *Machine Learning*

Nicolas Vayatis

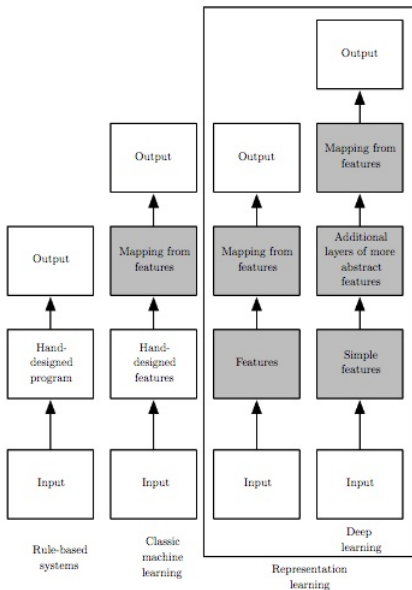
Janvier 2020

Session 3 - Représentation des données

What we have seen so far

- Machine Learning is about learning (= choosing = estimating) a function from data
- The key concept is the complexity of the function space ("hypothesis space") where we look for our solution ("how many functions we select from")
- The art of learning is to use the data to adjust the complexity of the hypothesis space - while implicitly considering the *approximation error*.
- In the particular case of least square linear regression, complexity calibration can (also) be achieved by only selecting and using a small subset of the variables (the problem of variable selection).

Another "Big picture" of Learning



Objectives for this class

- Focus on **feature selection** and **feature learning**: learning ("finding" or "choosing") a representation of the data
(Sessions 1-6: focused on learning functions for prediction and on bounding their generalization/prediction error *for a given set of features ("representation")*)
- Develop new regularisation/machine learning formulations for other applications such as learning (= estimating the missing entries of) matrices - for example used in recommender systems
- We will also learn about some **optimization** approaches to solve machine learning formulations/methods (possibly nonconvex optimization problems): **Optimization is central for machine learning**

A primer on parsity

- Sparsity-inducing regression methods: LASSO
- Motivation in linear predictive models: relaxation of ℓ_0 constraint on number of independent variables used, namely from minimizing

$$\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_0$$

to minimizing

$$\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1$$

- Advantages: tractable computations, interpretable models
- Byproduct: sparsistency (i.e. how many, and which variables to use)

Application (today): Matrix completion
with (rank) Sparsity
("Netflix Recommendation Competition")

		Item			
		W	X	Y	Z
User	A		4.5	2.0	
	B	4.0		3.5	
	C		5.0		2.0
	D		3.5	4.0	1.0

Rating Matrix

Application (today): Matrix completion with (rank) Sparsity ("Netflix Recommendation Competition")

- Given a matrix M with missing values, find the matrix X with *minimal rank* (why? - see later today) which coincides with the available coefficients of M :

$$\min_X \{\text{rank}(X)\} \text{ subject to } X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$

where $\Omega = \{(i,j) : M_{ij} \text{ not missing}\}$.

- How to solve this difficult optimization problem? Why is it difficult?

Sparse Feature Selection and Learning

- A. Feature Selection: LASSO with optimization methods
- B. Feature Learning: PCA and variants
- C. Applications: matrix completion, sparse coding, compressed sensing

A. Feature selection: LASSO with optimization methods

The LASSO for linear models

From ℓ_0 to ℓ_1

- Consider the LASSO estimation (learning) method: for any $\lambda > 0$,

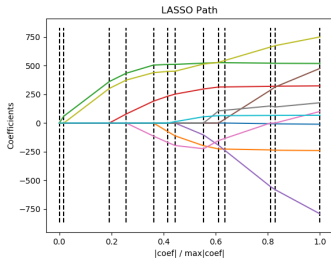
$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 \}$$

where the ℓ_1 -norm is:

$$\|\beta\|_1 = \sum_{j=1}^d |\beta_j|$$

Blessings of the LASSO

- Approximate solutions via efficient algorithms building the so-called *regularization path* (find for all values of λ the $\hat{\beta}(\lambda)$):



- Theoretical soundness: it can be shown that (if the real model is linear): as $n, d \rightarrow \infty$

$$\frac{1}{n} \mathbb{E}(\|\mathbf{x}\beta^* - \mathbf{x}\hat{\beta}\|^2) \leq C \|\beta^*\|_1 \sqrt{\frac{\log d}{n}}$$

Optimization methods for LASSO estimation

[mainly pointers to different approaches and literatures]

- Least Angle Regression
- Coordinate Descent
- Proximal methods

First algorithm: Least Angle Regression (LARS)

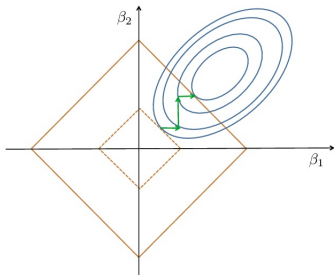
- LARS = variant of the incremental stagewise procedure for adding variables in a linear model
 - Least Angle Regression paper by Efron-Hastie-Johnstone-Tibshirani (AoS, 2004)
 - Previous work by Osborne et al. (2000) on the so-called homotopy method
 - Also related to greedy approaches such as Orthogonal Matching Pursuit (by Mallat, Zhang (1993), Mallat, Davis, Zhang (1994))
- Recovers the full regularization path $\lambda \rightarrow \hat{\beta}(\lambda)$ of the LASSO
- Success of the procedure based on the fact that LASSO path is piecewise linear.
- Computational efficiency: one ordinary least square computation at each step

Least Angle Regression: Pseudocode

- 1 Start with all coefficients β equal to zero.
- 2 Find the predictor x_j most correlated with y
- 3 Increase the coefficient β_j in the direction of the sign of its correlation with y until some other predictor x_k has as much correlation with $r = y - \hat{y}$ as x_j has.
- 4 Increase (β_j, β_k) in their joint least squares direction, until some other predictor x_m has as much correlation with the residual r .
- 5 Continue until: all predictors are in the model (corresponding to the solution when λ is small)

Second algorithm: Coordinate Descent

- Simple idea of one dimensional optimization with cyclic iteration over all variables, until convergence
- Optimization at each step amounts to a one-dimensional LASSO problem
- Solution obtained as a soft thresholding of the one-dimensional ordinary least square estimate.



Third algorithm: Proximal methods

- Parikh-Boyd tutorial paper (2013): "Much like Newton's method is a standard tool for solving unconstrained smooth optimization problems of modest size, proximal algorithms can be viewed as an analogous tool for nonsmooth, constrained, large-scale, or distributed versions of these problems."
- Early work goes back to Moreau (1960s) then Nemirovski, Yudin (1983)
- Rediscovered around 2005 with applications to signal processing and solving certain optimization problems

Proximal method (1/4)

Principle

- Applies to a problem of the form:

$$\min_{\beta} \{L(\beta) + \psi(\beta)\}$$

when: L is smooth, convex, with "bounded" gradient, and ψ is continuous, convex, but non-smooth

- The proximal algorithm is a descent algorithm which provides a sequence β_t obtained as follows: at each step t ,

$$\beta_t = \text{prox}(\psi, \beta_{t-1} - \nabla L(\beta_{t-1}))$$

where prox is the so-called proximal operator (generalizes the concept of orthogonal projection)

Proximal method (2/4)

Definition of proximal operator

- Definition of the proximal operator for the nonsmooth term ψ of the objective $L + \psi$

$$\text{prox}(\psi, z) = \arg \min_{\beta} \left\{ \frac{1}{2} \|\beta - z\|_2^2 + \psi(\beta) \right\}$$

- Interpretation: The proximal operator finds a point that corresponds to a trade-off between minimizing ψ and being near to the point z .

Proximal method (3/4)

Application to LASSO

- Here: $L(\beta) = \frac{1}{2}\|X\beta - y\|_2^2$ and $\psi(\beta) = \lambda\|\beta\|_1$
- Gradient step relies on the gradient of the smooth term L :

$$\nabla L(\beta) = X^T(X\beta - y)$$

- Proximal operator for the ℓ_1 norm is given by:

$$\text{prox}(\lambda\|\cdot\|_1, z) = (z - \lambda)_+ - (-z - \lambda)_+$$

(soft thresholding operator on each component of z)

- Also called ISTA (for Iterative Shrinkage Thresholding Algorithm)

Proximal methods (4/4)

Discussion

- Special cases: gradient descent, projected gradient
- Accelerated version: FISTA for Fast Iterative Shrinkage Thresholding Algorithm
- Numerical convergence: from $O(1/t)$ to $O(1/t^2)$

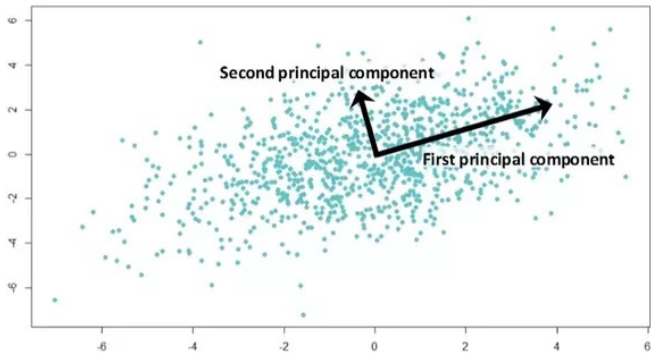
B. Feature Learning: PCA and variants

What all students should know

PCA

- Motivation: Dimensionality reduction
- Principle: Find an orthogonal basis to represent (project on) the data, which captures the directions of highest dispersion (variance) of the data
- Underlying assumption: Gaussian, highly correlated data

Idea of PCA



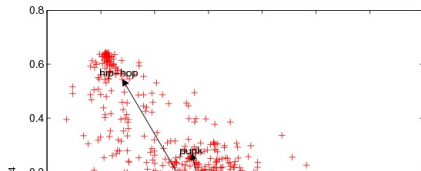
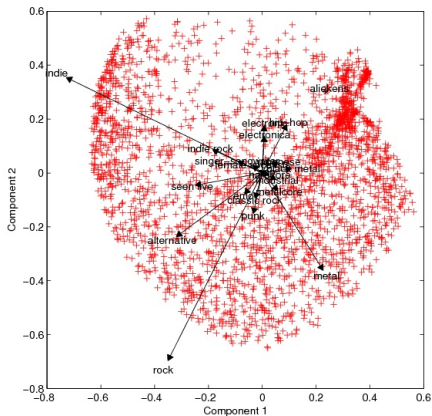
PCA

Classical construction

- Compute the covariance (or correlation) matrix of the data
- Find the eigen-elements (values/vectors) - eigenvectors being orthogonal - of this matrix
- Principal components are ordered from the larger eigenvalue to the smallest
- Dimensionality reduction from d to (small) r is performed by projecting the initial data points on the first (principal) r eigenvectors

PCA applied to music recommendation

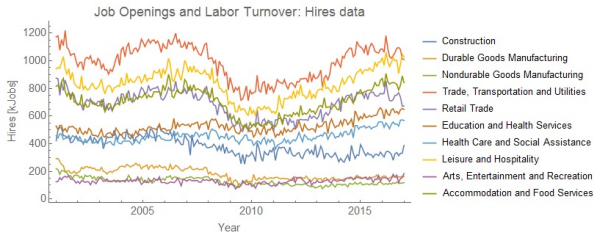
LastFM data set



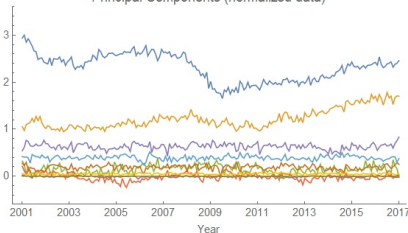
PCA applied to time series

Job hiring data

JOLTS data set available at <https://www.bls.gov/jltt/>



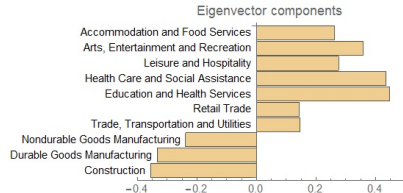
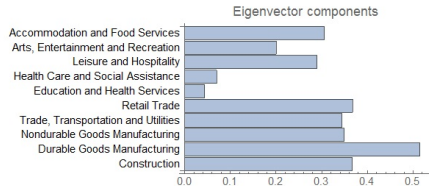
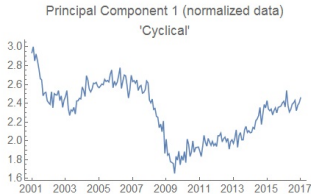
Principal Components (normalized data)



PCA applied to time series

Job hiring data

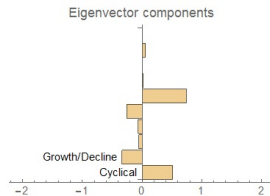
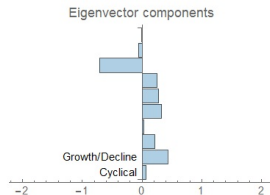
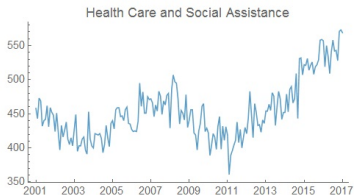
Components interpretation



PCA applied to time series

Job hiring data

Projection on principal components



PCA applied to time series

Financial data (1/2)

Paper by Avellenada and Lee (2008)

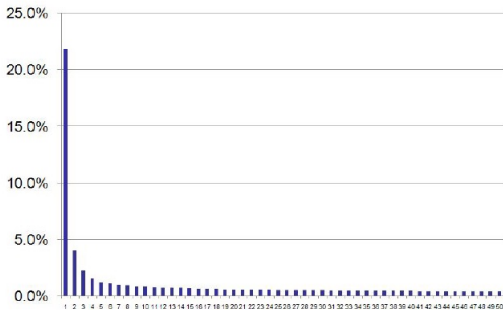


Figure 1: Eigenvalues of the correlation matrix of market returns computed on May 1 2007 estimated using a 1-year window (measured as percentage of explained variance)

PCA applied to time series Financial data (2/2)

Paper by Avellenada and Lee (2008)

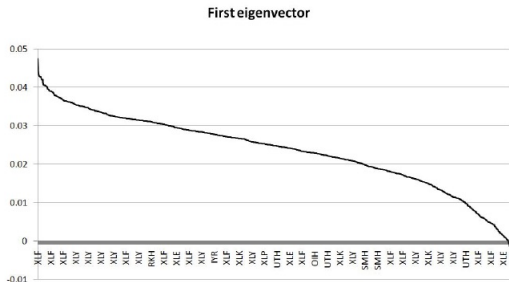


Figure 4: First eigenvector sorted by coefficient size. The x-axis shows the ETF corresponding to the industry sector of each stock.

A different view on PCA

- Denote by X the data matrix of size $d \times n$ (assume that the points are centered) and by $\|M\|_F^2 = \sum_{i,j} M_{ij}^2$ the square of the *Frobenius norm* of the matrix $M = (M_{ij})_{ij}$
- Solve the minimization problem:

$$\min_{P,Z} \|X - PZ\|_F^2 \text{ subject to } P^T P = I_r$$

where P is the projection matrix of size $d \times r$ (the matrix whose columns are the first r eigenvectors), and Z is $r \times n$ matrix of the projected points in the r -dimensional subspace. We also have the *orthogonality* constraint $P^T P = I_r$ (eigenvectors are orthogonal)

A low-rank formulation of PCA

- An alternative formulation to the previous optimization problem, by setting: $A = PZ$, is:

$$\min_A \|X - A\|_F^2 \text{ subject to } \text{rank}(A) = r$$

- Theoretical result (Vidal, Ma, Sastry (2016)): an optimal solution to this problem is given by:

$$A = U_r \Sigma_r V_r$$

where U_r and V_r have orthogonal columns of size $d \times r$ and $n \times r$ respectively, Σ_r diagonal square matrix of size $r \times r$. The matrices U_r , Σ_r , V_r correspond to the **reduced singular value decomposition (SVD) of matrix X** .

Some linear algebra background: SVD decomposition

A generalization of eigenvalues and eigenvectors.

- Definition: σ is a singular value of a rectangular $d \times n$ matrix X if there exist unit two vectors $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^n$ such that

$$X^T u = \sigma v \quad \text{and} \quad Xv = \sigma u$$

The vectors u and v are called **singular vectors**.

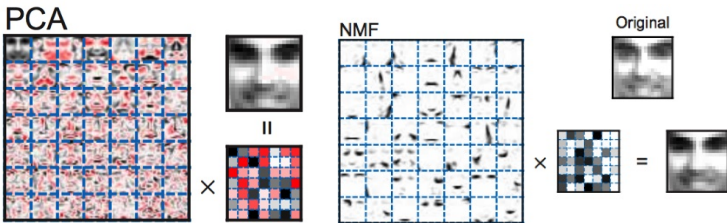
- Theorem: For any rectangular matrix, there exist U and V orthogonal matrices of size $d \times d$ and $n \times n$ respectively and a diagonal matrix Σ of size $d \times n$ such that:

$$X = U\Sigma V^T$$

Some issues with PCA

- PCA is sensitive to outliers; empirical covariance matrix converges to real covariance slowly wrt sample size...
- What if natural components are not Gaussian? what if they are not orthogonal but independent (check more than just their correlation)? ...
- What about interpretation? Maybe we need nonnegativity of matrix Z (the new data representation) \rightarrow Nonnegative Matrix Factorization

Nonnegative Matrix Factorization



D.D. Lee and H. S.Seung, "Learning the parts of objects by non-negative matrix factorization", Nature 401 (6755), pp. 788–791, 1999

PCA Generalisations: Example Machine Learning Formulation

- Example: Robust PCA by Candès, Li, Ma, Wright (2011)
- Motivation: assume a decomposition of the data matrix $X = L + S$ where L is low rank and S is sparse.
- *Principal Component Pursuit*: the *nuclear norm* (also called *Trace norm*) $\|\cdot\|_*$ defined as the sum of singular values; note with $\|\cdot\|_1$ the ℓ_1 matrix norm (sum of the absolute values of all the entries of the matrix). We search for matrices L and S :

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \text{ subject to } L + S = X$$

- Main theoretical result: under some assumptions the *exact* solution may be recovered by this procedure

Other variants of PCA

- Sparse PCA
- Nonlinear PCA, Kernel PCA
- Multidimensional scaling, Local embeddings, Laplacian eigenmaps
- ...

Reference: book by Vidal, Ma, Sastry. Generalized Principal Component Analysis. Springer (2016)

C. Applications: matrix completion, compressed sensing

Matrix completion: Recommender Systems Application

	Item			
	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

User Matrix

X

	W	X	Y	Z
	1.5	1.2	1.0	0.8
	1.7	0.6	1.1	0.4

Item Matrix

Matrix completion: Problem statement

- Original optimization formulation (kind of "Ivanov Regularization" with no error on the available matrix entries - our data)

$$\min_X \{\text{rank}(X)\} \text{ subject to } X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$

where $\Omega = \{(i,j) : M_{ij} \text{ the available data}\}$.

- **Key Challenge:** Non-convex problem, hard to solve

Matrix completion: Convex Relaxation

- Recall the *nuclear norm* of X is $\|X\|_* = \sum_{i=1}^{\min(n,m)} \sigma_i$, where σ_i are the singular values of X (recall the SVD of X is $X = U\Sigma V^T$)
- Convex formulation of the matrix completion problem:

$$\min_X \|X\|_* \text{ subject to } X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$

where $\Omega = \{(i,j) : M_{ij} \text{ the available data}\}$.

- Regularization formulation:** Nuclear norm penalty

$$\min_X \left\{ \frac{1}{2} \sum_{ij \in \Omega} (X_{ij} - M_{ij})^2 + \lambda \|X\|_* \right\}$$

Matrix completion Solution (1/2)

- Simplified problem (no mask Ω):

$$\min_X \left\{ \frac{1}{2} \|X - M\|^2 + \lambda \|X\|_* \right\}$$

- The solution is closed form and given by:

$$\text{shrink}(X, \lambda) = U \Sigma(\lambda) V^T$$

where $\Sigma(\lambda) = \text{diag}((\sigma_i - \lambda)_+)$

- Note: the solution uses only the singular values that are larger than λ ...

Matrix completion Solution (2/2)

- Need a trick to deal with the Ω
- Use an auxiliary matrix Y which is complete
- Define $\Pi_{\Omega}(X)$ the matrix with coefficients X_{ij} if $(i,j) \in \Omega$ and zero if $(i,j) \notin \Omega$
- Iterative algorithm (called "SVT"):
 - ① Set $\lambda > 0$ and sequence of step sizes $(\delta_k)_{k \geq 1}$
 - ② Start with $Y_0 = 0$ matrix of size $n \times m$
 - ③ At each step k , compute:

$$\begin{cases} X_k &= \text{shrink}(Y_{k-1}, \lambda) \\ Y_k &= Y_{k-1} + \delta_k \Pi_{\Omega}(M - X_k) \end{cases}$$

C2. Dictionary learning

Motivations and references

- Some features (to represent the data) may be good for compression but not for interpretation (and vice versa); they may also simply fail to "lead to" sparse representations (e.g., learn functions that use only a few of the features)
- Can we learn data features (representation) so that the functions we learn (estimate) in that representation ("space") are also sparse?
- Idea is to exploit the fact that *similar patterns may be repeated in the data (even if they are not smooth)*
- (Can also be used to handle some cases of non-stationarity)

References: Olshausen and Field (1997) Kreutz-Delgado et al. (2003), Mairal, Elad, Sapiro (2008), Gribonval et al. (2015)

Sparse coding Formulation

- Objective: find both A (the "features") and Y that yield to the sparse representation of the data X up to some error ε
- Formulation:

$$\min_{A, Y} \left\{ \sum_{i=1}^n \|Y_i\|_0 \right\} \text{ subject to } \|X - AY\|_2 \leq \varepsilon$$

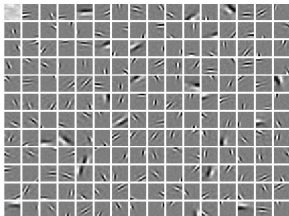
Sparse coding

Towards nonconvex optimization

- Same complexity as ℓ_0 norm minimization problem. In practice, it is solved with an ℓ_1 -type relaxation
- But: for fixed A , minimization over Y is convex but the joint optimization wrt both A and Y is not convex
- Main strategy for non convex matrix factorization problems: alternating minimization (Douglas-Rachford) or Block coordinate descent

Sparse coding: Examples

- Images (text? multimedia?, etc)



- Representation of consumer products ("meta-attributes") and utility functions (see also conjoint analysis and Multi-task Learning in Sessions 13-14).

Sparsity

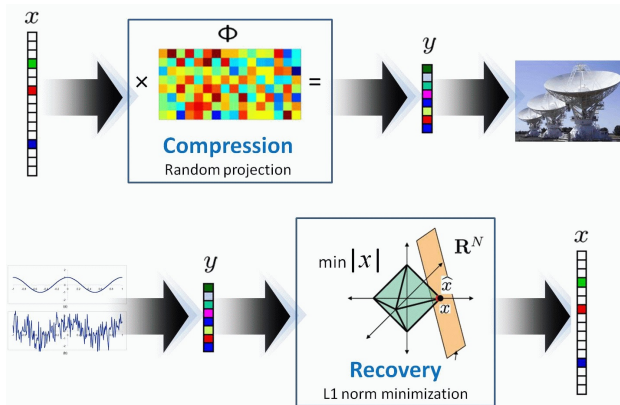
C.3. Compressed sensing

A revolution in signal processing

- Classical signal representation relies on first measuring then compressing (the information/data - hence "finding the rules/laws")
- Take-home message: Sparsity and regularization are the keys for extreme compression
- Technological breakthroughs have been achieved in imaging such as the "one-pixel camera"
- Pioneering work by Candès-Romberg-Tao (2006) and Donoho (2006)

Data compression Old and New

Compressed sensing



Classical compression

Compressed Sensing Setup

- Want to recover the signal $y \in \mathbb{R}^d$ based on few measurements $x_i = z_i^T y$ for $i = 1, \dots, n$ with $n \ll d$ where z_i are random "directions".
- Assumption: the signal y has a sparse linear representation, meaning that there exists a sparse vector β such that $y = \Psi\beta$ where Ψ is the matrix of basis vectors.

Compressed Sensing Optimization problem

- Compressed sensing can then be formulated as a linear program wrt β :

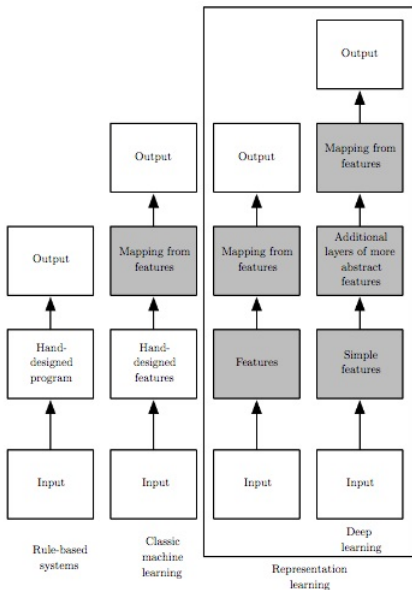
$$\min_{\beta \in \mathbb{R}^d} \|\beta\|_1 \text{ subject to } X = Z\Psi\beta$$

where the vector $X \in \mathbb{R}^n$ contains the observations, and the two matrices Z (design matrix of size $n \times d$) and Ψ (square matrix $d \times d$, basis of \mathbb{R}^d) are fixed and known.

- Eventually, the signal is recovered (de-compressed) thanks to the relation $y = \Psi\beta$.

Remark: there is a family of procedures depending on the choice of the design matrix (usually random matrix with gaussian or Rademacher entries).

Another "Big picture" of Learning



Wrap-up and other topics

- Representation learning aims at extracting structure from complex low-level data
- Practical methods rely on high dimensional statistical modeling, linear algebra and optimization formulations inspired from machine learning techniques
- Dictionary learning is an example of unsupervised learning task
- Other unsupervised learning problems are:
 - Clustering (or segmentation or unsupervised classification)
 - Anomaly detection
 - Novelty detection