INSEAD PhD Course

Foundations of Machine Learning and AI

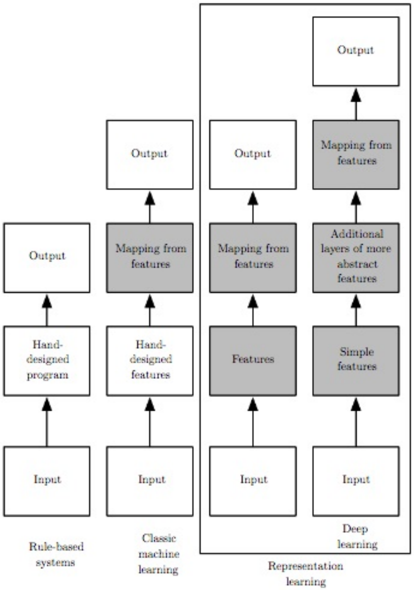
Theodoros Evgeniou - Nicolas Vayatis

Sessions 5-6: Data representations, feature learning and applications

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
- 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**

What we know about sparsity from sessions 3-4

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

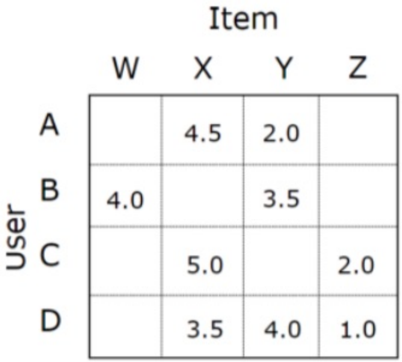
$$\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_0$$

  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

Feature Selection and Learning

A. Feature "Learning": PCA and variants
B. Feature Selection: LASSO with optimization methods
C. Applications: sparse coding, (kernels), matrix completion

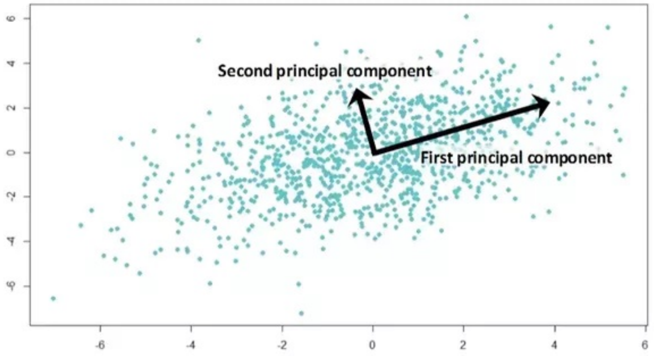
A. Feature "Learning": Principal Component Analysis (PCA) and variants

1. Classical PCA

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



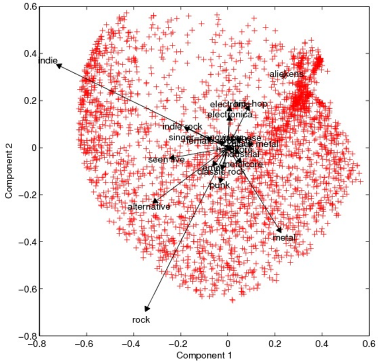
Second principal component
First principal component
6
4
2
0
-2
-4
-6

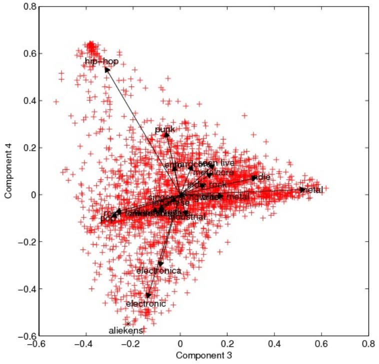
PCA
Classical construction

- 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

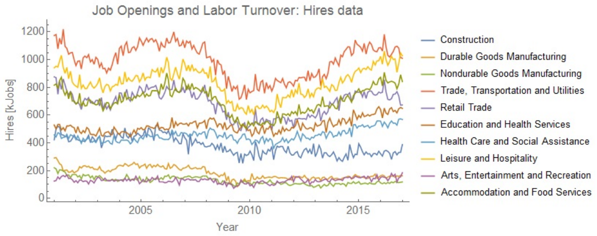
LastFM data set `http://anthony.liekens.net/index.php/Computers/DataMining`

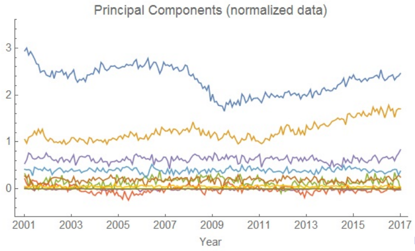




PCA applied to time series Job hiring data (1/3)

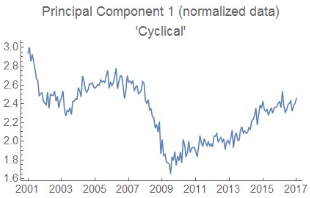
JOLTS data set available at `https://www.bls.gov/jlt/>`

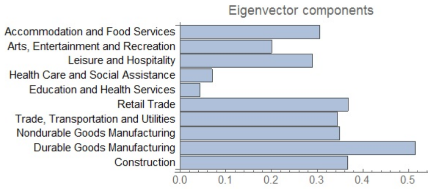


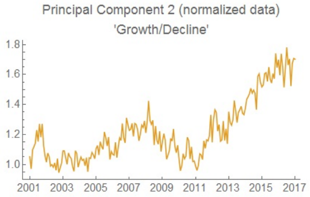


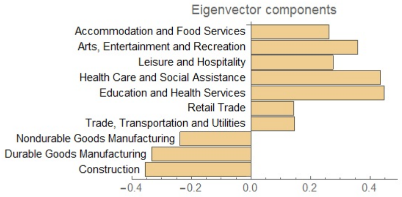
PCA applied to time series
Job hiring data (2/3)

Components interpretation



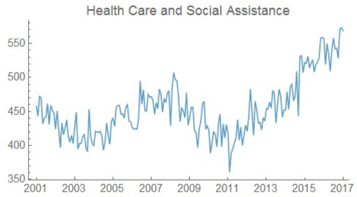


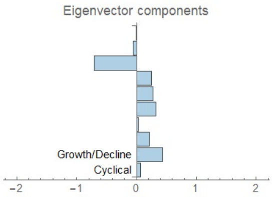


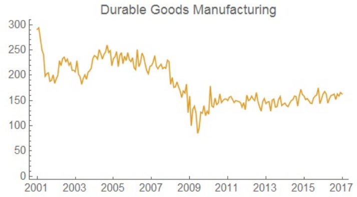


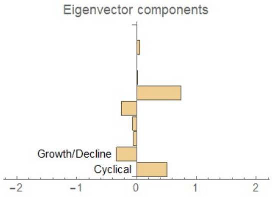
PCA applied to time series Job hiring data (3/3)

Projection on principal components









PCA applied to time series
Financial data (1/2)

Paper by Avellenada and Lee (2008)

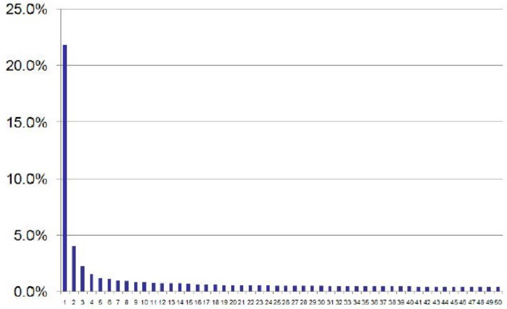


Figure 1: Eigenvalues of the correlation matrix of market returns computed on May 1 2007 estimated using a 1-year window (measured as percentage of explained variance)

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

Paper by Avellenada and Lee (2008)

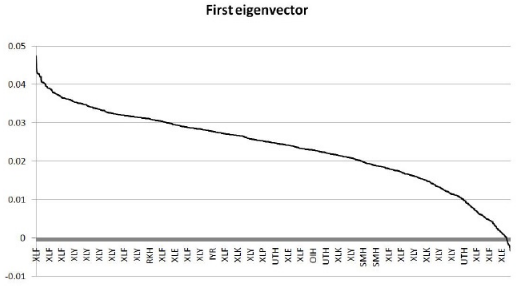


Figure 4: First eigenvector sorted by coefficient size. The x-axis shows the ETF corresponding to the industry sector of each stock.

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:

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

- 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

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

  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:

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

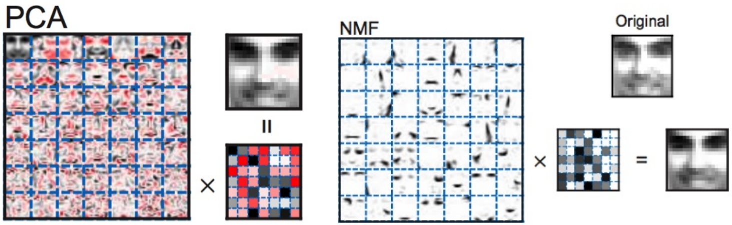
A. Feature "Learning":
Principal Component Analysis (PCA) and variants

2. Nonnegative Matrix Factorization (NMF)

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

A. Feature "Learning": Principal Component Analysis (PCA) and variants

3. Robust PCA

A Machine Learning-type Formulation
Robust PCA

- Introduced 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)

B. 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 $\lambda > 0$,

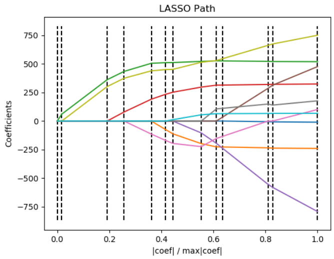
$$\widehat{\beta}_\lambda \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^d} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 \right\}$$

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 $\widehat{\beta}(\lambda)$):



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

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

Optimization methods for LASSO estimation

[mainly pointers to different approaches and literature]

- Least Angle Regression
- Coordinate Descent
- Proximal methods

Optional material

Optimization methods applied to LASSO

First algorithm: Least Angle Regression (LARS)

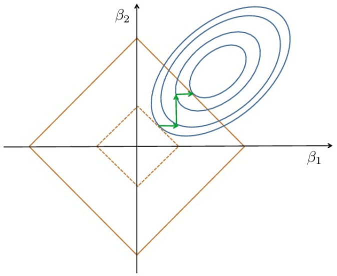
- LARS = variant of the incremental stagewise procedure for adding variables in a linear model
  - Least Angle Regression paper by Efron-Hastie-Johnstone-Tibshirani (AoS, 2004)
  - Previous work by Osborne et al. (2000) on the so-called homotopy method
  - Also related to greedy approaches such as Orthogonal Matching Pursuit (by Mallat, Zhang (1993), Mallat, Davis, Zhang (1994))
- Recovers the full regularization path $\lambda \to \hat{\beta}(\lambda)$ 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
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,

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

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

Proximal method (1/2)
Definition of proximal operator

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

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

- 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 (2/2)
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:

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

  (soft thresholding operator on each component of z)
- Also called ISTA (for Iterative Shrinkage Thresholding Algorithm)

End of optional material

C. Applications:

1. Sparse coding

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 convolutional coding

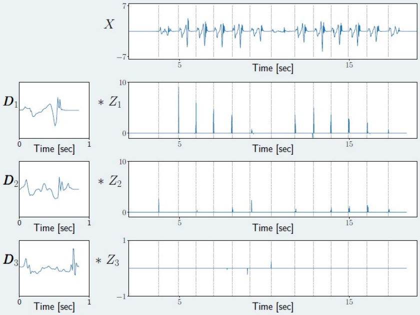
Notation

- X is a signal of length T
- $\mathcal{E}$ is a noise signal of length T
- $\boldsymbol{D}$ is a set of K patterns of length W
- Z is a signal of length $L = T - W + 1$ in $\mathbb{R}^K$

Sparse Convolutional model:

$$X[t] = \sum_{k=1}^{K} (\boldsymbol{D}_k * Z_k)[t] + \mathcal{E}[t]$$

with Z sparse. Few of its coefficients are non-zero.



Sparse (linear) coding

Notation

- x is a vector in $\mathbb{R}^T$
- ϵ is a noise vector in $\mathbb{R}^T$
- D is a matrix in $\mathbb{R}^{T\times LK}$
- z is a coding vector in $\mathbb{R}^{LK}$

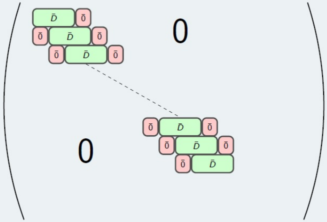
Sparse Linear model:

$$x = Dz + \epsilon$$

with z sparse. Few of its coefficients are non-zero.

Link with convolutional model

$D =$



Sparse coding
Formulation

- Objective: find both the features D and the activations Z that yield to the sparse representation of the data X up to some error ε
- Formulation:

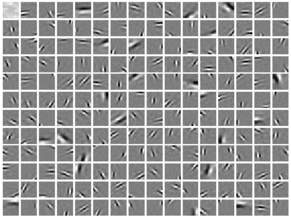
$$\min_{D,Z} \left\{ \sum_{i=1}^{n} \|Z_i\|_0 \right\} \text{ subject to } \|X - DZ\|_2 \leq \varepsilon$$

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 D, minimization over Z is convex while the joint optimization wrt both Z and Z 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)

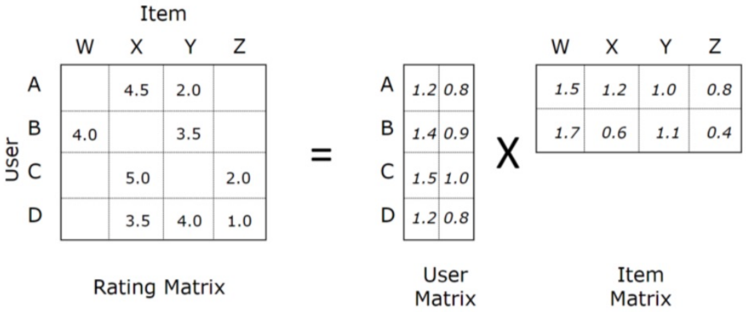


- Representation of consumer products (”meta-attributes”) and utility functions (see also conjoint analysis and Multi-task Learning in Sessions 13-14).

C. Applications

2. Matrix completion

Matrix completion: Recommender Systems Application



$$\begin{array}{c|cccc} & W & X & Y & Z \\ \hline A & & 4.5 & 2.0 & \\ B & 4.0 & & 3.5 & \\ C & & 5.0 & & 2.0 \\ D & & 3.5 & 4.0 & 1.0 \end{array} = \begin{array}{c|cc} A & 1.2 & 0.8 \\ B & 1.4 & 0.9 \\ C & 1.5 & 1.0 \\ D & 1.2 & 0.8 \end{array} \times \begin{array}{cccc} W & X & Y & Z \\ \hline 1.5 & 1.2 & 1.0 & 0.8 \\ 1.7 & 0.6 & 1.1 & 0.4 \end{array}$$

Matrix completion: Problem statement

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

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

  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:

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

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

- **Regularization formulation**: Nuclear norm penalty

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

Optional material

Resolution of the matrix completion problem

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:

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

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

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

Matrix completion
Solution (2/2)

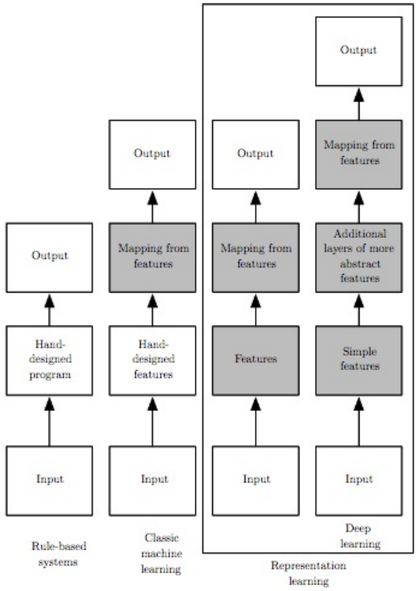
- Need a trick to deal with the Ω
- Use an auxiliary matrix Y which is complete
- Define $\Pi_\Omega(X)$ the matrix with coefficients X_{ij} if $(i,j) \in \Omega$ and zero if $(i,j) \notin \Omega$
- Iterative algorithm (called "SVT"):
  1. Set $\lambda > 0$ and sequence of step sizes $(\delta_k)_{k\geq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

$$\left\{ \begin{array}{ll} X_k & = \mathrm{shrink}(Y_{k-1}, \lambda) \\ Y_k & = Y_{k-1} + \delta_k \Pi_\Omega(M - X_k) \end{array} \right.$$

End of optional material

Coming next

Another "Big picture" of Learning



Next sessions

- Deep Learning (also learn "hierarchical representations" of the data - what if "nature" has this structure?)
- More on optimization (e.g., stochastic gradient)
- Q&A, catch up, class projects
- Overview of popular algorithms (sessions 9-10)