Sparsity in Machine Learning

Sparsifiers
SVD
Linear Regression
K-means

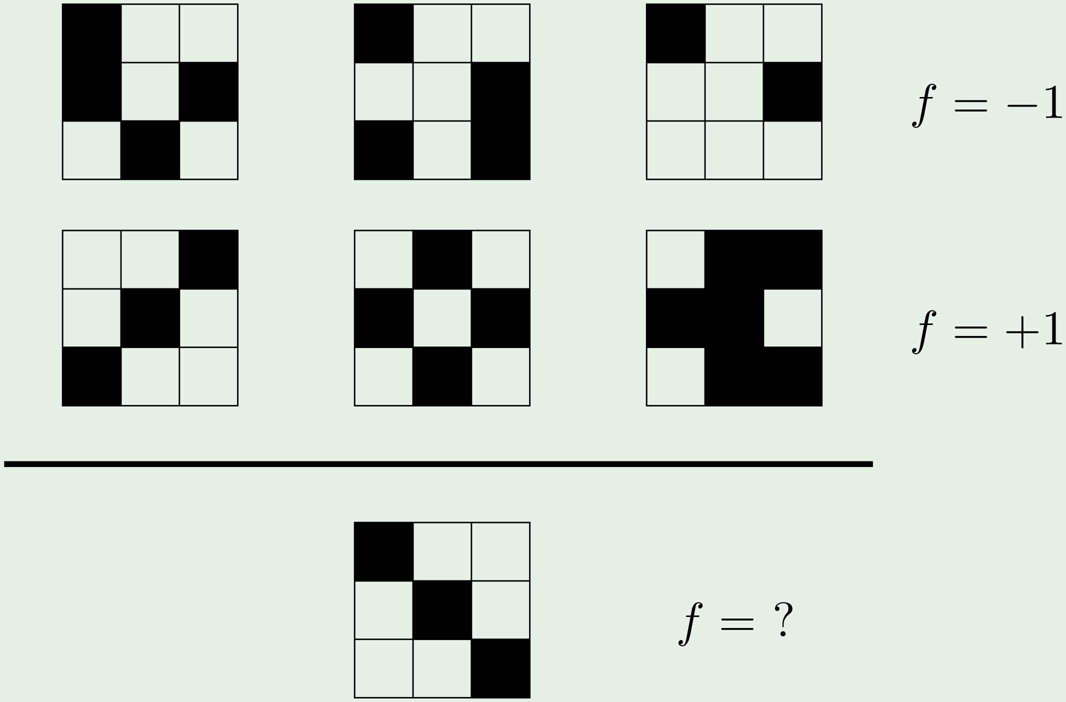
M. Magdon-Ismail
Rensselaer Polytechnic Institute

Rensselaer

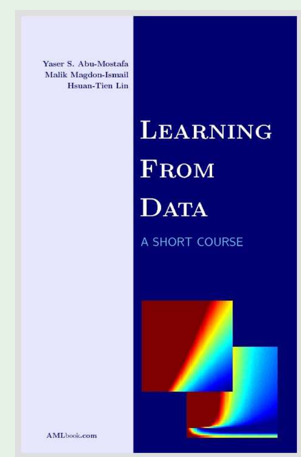
(Joint work with C. Boutsidis and P. Drineas)

June 20, 2013.

Out-of-Sample Prediction



- A **pattern** exists (f)
- We **don't know it**
- We **have data** to learn it ($\mathcal{D}$)



Data

$$\underbrace{n \text{ data points}}\begin{bmatrix} \text{---}\ \mathbf{x}_1^{\mathrm{T}}\ \text{---} \\ \text{---}\ \mathbf{x}_2^{\mathrm{T}}\ \text{---} \\ \vdots \\ \text{---}\ \mathbf{x}_n^{\mathrm{T}}\ \text{---} \end{bmatrix} = \overbrace{\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}}^{d \text{ dimensions}} \qquad \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} \\ \mathbf{y}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{y}_n^{\mathrm{T}} \end{bmatrix}$$

$$\mathrm{X} \in \mathbb{R}^{n\times d} \qquad\qquad \mathrm{Y} \in \mathbb{R}^{n\times \omega}$$

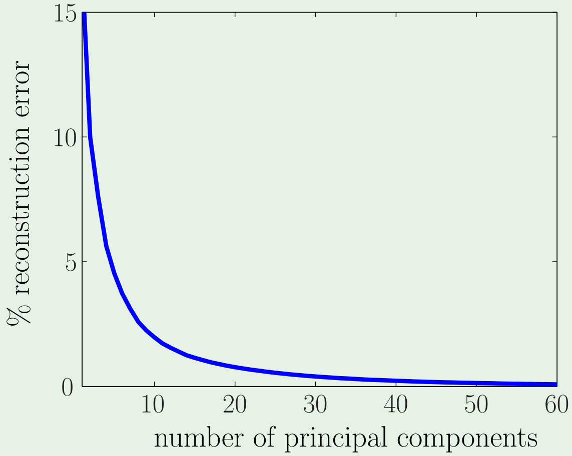
viewers $\times$ movie ratings

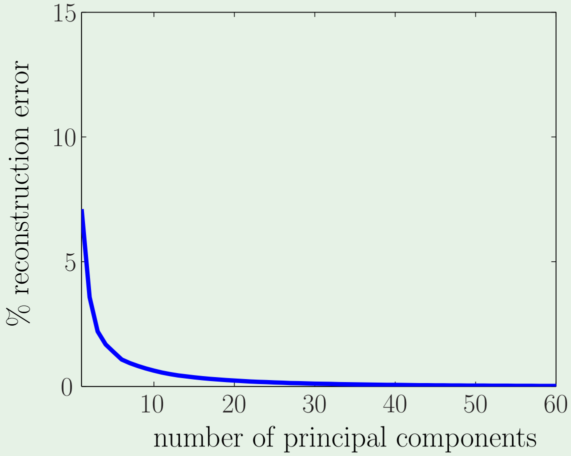
credit applicants $\times$ credit features $\qquad y = \pm 1$ (approve or not)

Data

$X \in \mathbb{R}^{231\times 174}$

$Y \in \mathbb{R}^{231\times 166}$





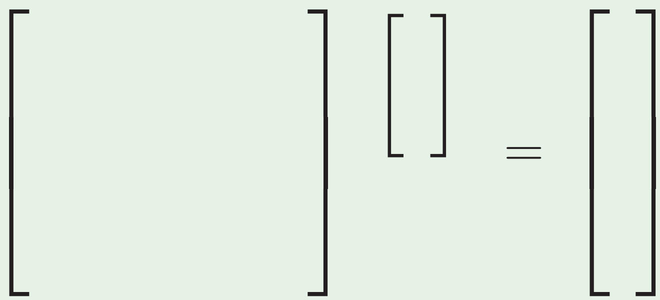
Sparsity

Represent your solution using **only a few ...**

Sparsity

Represent your solution using **only a few ...**

Example: linear regression



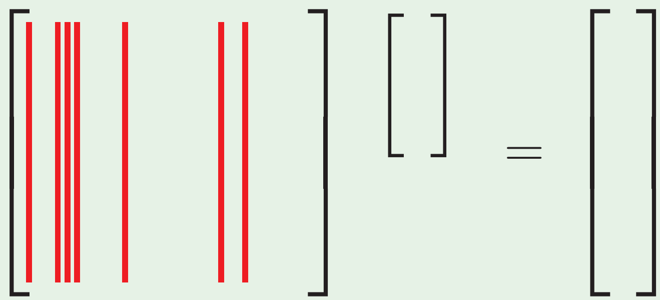
$$X\mathbf{w} = \mathbf{y}$$

$\mathbf{y}$ is an optimal linear combination of the columns in X.

Sparsity

Represent your solution using **only a few . . .**

Example: linear regression



$$X\mathbf{w} = \mathbf{y}$$

$\mathbf{y}$ is an optimal linear combination of **only a few** columns in X.

(sparse regression; regularization ($\| \mathbf{w} \|_0 \le k$); feature subset selection; . . .)

Sparsity is Good

Sparse solutions generalize to out-of-sample better.

Sparse solutions are easier to interpret.

Computations are more efficient.

Problem: sparsity is a combinatorial constraint.

Singular Value Decomposition (SVD)

$$\mathrm{X} = \begin{bmatrix} \mathrm{U}_k & \mathrm{U}_{d-k} \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{d-k} \end{bmatrix} \begin{bmatrix} \mathrm{V}_k^{\mathrm{T}} \\ \mathrm{V}_{d-k}^{\mathrm{T}} \end{bmatrix} \qquad O(nd^2)$$

$$\underset{(n \times d)}{\mathrm{U}} \qquad \underset{(d \times d)}{\Sigma} \qquad \underset{(d \times d)}{\mathrm{V}^{\mathrm{T}}}$$

$$\begin{aligned} \mathrm{X}_k &= \mathrm{U}_k \Sigma_k \mathrm{V}_k^{\mathrm{T}} \\ &= \mathrm{X}\mathrm{V}_k \mathrm{V}_k^{\mathrm{T}} \end{aligned}$$

X_k is the best rank-k approximation to X.

Reconstruction of X using **only a few** *features*.

X X_{20} X_{40} X_{60}

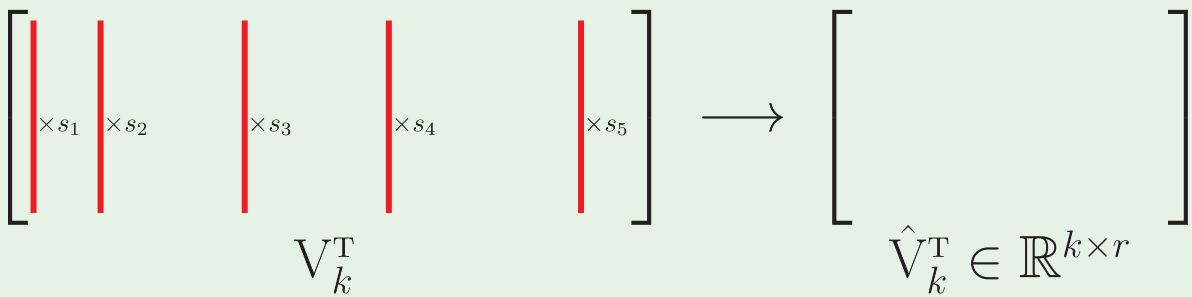
V_k is an orthonormal basis for the best k-dimensional subspace of the row space of X.

V_k and Sparsity

Principal Components Analysis (PCA):

$$\underset{(n \times k)}{Z} = XV_k$$

Feature subset selection: Important "dimensions" of V_k^T are important for X



The sampled r columns are "good" if

$$I = V_k^T V_k \approx \hat{V}_k^T \hat{V}_k.$$

Sampling schemes: Largest norm (Jollife, 1972);
Randomized norm sampling (Rudelson, 1999; RudelsonVershynin, 2007);
Greedy (Batson et al, 2009; **BDM, 2011**).

Approximate SVD

$$\mathrm{X} = \underbrace{\mathrm{X}\mathrm{V}_k\mathrm{V}_k^{\mathrm{T}}}_{\mathrm{X}_k} + \mathrm{E}$$

Let $\hat{\mathrm{V}}_k$ be an approximate V_k

$$\mathrm{X} = \mathrm{X}\hat{\mathrm{V}}_k\hat{\mathrm{V}}_k^{\mathrm{T}} + \hat{\mathrm{E}}$$

$\hat{\mathrm{V}}_k$ is good if

$$\|\,\hat{\mathrm{E}}\,\| \le (1+\epsilon)\|\,\mathrm{E}\,\|.$$

Approximate SVD

$Z = XR$

$Q = \text{QR.FACTORIZE}(Z)$

$\hat{V}_k \leftarrow \text{SVD}_k(Q^{T}X)$

$R \sim \mathcal{N}(d \times r), \; Z \in \mathbb{R}^{n \times r}$

Theorem. Let $r = \left\lceil k(1+\frac{1}{\epsilon}) \right\rceil$ and $E = X - X\hat{V}_k\hat{V}_k^{T}$. Then,

$$\mathbb{E}\left[\| E \|\right] \leq (1+\epsilon)\| X - X_k \|$$

running time is $O(ndk) = o(\text{SVD})$

[BDM, FOCS 2011]

Approximate SVD

$k = 20$ $\quad$ $k = 40$ $\quad$ $k = 60$

Exact SVD

Approx. SVD

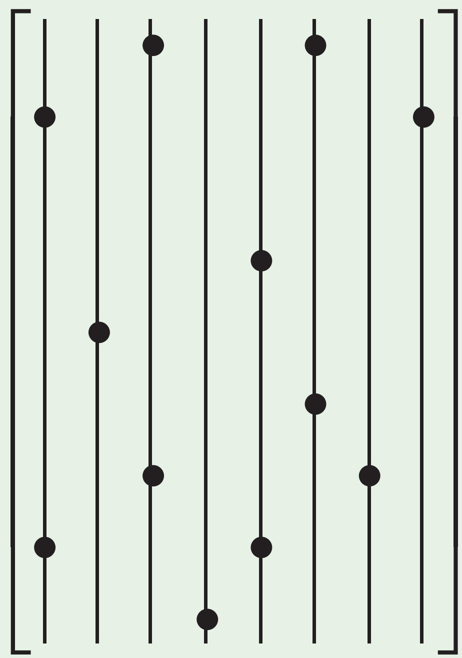
Sparse PCA

A principal component is a "dense" combination of the feature dimensions.

A sparse principal component is a combination of a few feature dimensions.

Want V_k to have a few non-zeros in each column

$\mathrm{V}_k =$



Sparse PCA

1: Choose a few columns C of X; $C \in \mathbb{R}^{n\times r}$.

2: Find the best rank-k approximation of X in the span of C, $X_{C,k}$.

3: Compute the SVD_k of $X_{C,k}$:

$$X_{C,k} = U_{C,k}\Sigma_{C,k}V_{C,k}^{T}.$$

4:

$$Z = X_{C,k}V_{C,k} = U_{C,k}\Sigma_{C,k}.$$

Each feature in Z is a mixture of **only the few** original r feature dimensions in C.

$$\| X - ZZ^{\dagger}X \| \le \| X - ZV_{C,k}^{T} \| = \| X - X_{C,k} \|.$$

Sparse PCA

1: Choose a few columns C of X; $\mathrm{C} \in \mathbb{R}^{n\times r}$.
2: Find the best rank-k approximation of X in the span of C, $\mathrm{X}_{\mathrm{C},k}$.
3: Compute the SVD_k of

$$\mathrm{X}_{\mathrm{C},k} = \mathrm{U}_{\mathrm{C},k}\Sigma_{\mathrm{C},k}\mathrm{V}_{\mathrm{C},k}^{\mathrm{T}}.$$

4:

$$\mathrm{Z} = \mathrm{X}_{\mathrm{C},k}\mathrm{V}_{\mathrm{C},k}.$$

Each feature in Z is a mixture of **only the few** original r feature dimensions in C.

$$\|\,\mathrm{X} - \mathrm{Z}\mathrm{Z}^{\dagger}\mathrm{X}\,\| \leq \|\,\mathrm{X} - \mathrm{Z}\mathrm{V}_{\mathrm{C},k}^{\mathrm{T}}\,\| = \|\,\mathrm{X} - \mathrm{X}_{\mathrm{C},k}\,\| \leq \left(1 + O(\tfrac{2k}{r})\right)\|\,\mathrm{X} - \mathrm{X}_k\,\|.$$

[BDM, FOCS 2011]

Sparse PCA

$k = 20$ $k = 40$ $k = 60$

Dense PCA

Sparse PCA, $r = 2k$

Theorem. One can construct, in $o(\textsc{svd})$, sparse features that are as good as exact dense PCA-features.

Feature Subset Selection: K-Means

Choose a few features

Cluster the data using these features

PCA - dense features.

Sparse features: feature subset selection.

Compare the clusterings on all the dimensions.

Feature Subset Selection: K-Means

Full | PCA, $k = 20$ | Sparse, $r = 2k$

3 clusters

4 Clusters

Theorem. There is a subset of features of size $O(\#clusters)$ which produces nearly the optimal partition (within a constant factor). One can quickly produce features with a log-approximation factor.

[BDM,2013]

Feature Subset Selection: Regression

$$\begin{bmatrix} \\ \\ \end{bmatrix} =$$

$$\mathrm{X} \quad \mathbf{w} = \mathrm{Y}$$

$$\hat{\mathrm{Y}}$$

Feature Subset Selection: Regression

PCA | Sparse, $r = 2k$

$k = 20$

$k = 40$

Theorem. There are $O(k)$ pure features which performs as well regressing on PCA_k features (to within small additive error).

[BDM,2013]

The Proofs

All the algorithms use the sparsifier of V_k^T in [BDM,FOCS2011].

1. Choose columns of V_k^T to preserve its singular values.

2. Ensure that the selected columns preserve the structural properties of the objective with respect to the columns of X that are sampled.

(In all cases, the objective is a squared (Frobenius) error.)

THANKS!

Focussed on columns of V_k^T to "sparsify" dimensions.

Can quickly approximate V_k.

Can efficiently use it to obtain

- sparse PCA
- small subset of features for k-means, which results in near optimal clustering.
- small subset of features for regression, which results regression comparable to PCA_k.

Sparse solutions: easy to interpret; better generalizers; faster computations.

Using U_k instead of V_k^T one can "sparsify" data points to get coresets. [BDM,2013]