Sparse Learning

Jun Zhu
dcszj@mail.tsinghua.edu.cn
http://bigml.cs.tsinghua.edu.cn/~jun
State Key Lab of Intelligent Technology & Systems
Tsinghua University

May 26, 2015

Outline

- Sparse learning
  - Sparse learning on vectors
  - Sparse learning on matrices
  - Dictionary learning

Two Important Aspects

- Model goodness:
  - Often defined in terms of prediction accuracy

- Model parsimony:
  - Simpler models are preferred for the sake of scientific insight into the x – y relationship

Example – diabetes study

- 442 diabetes patients were measured on 10 baseline variables; a prediction model was desired for the response variable, a measure of disease progression one year after baseline

	AGE	SEX	BMI	BP	Serum measurements						Response
Patient	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y
1	59	2	32.1	101	157	93.2	38	4	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4	4.7	85	141
4	24	1	25.3	84	198	131.4	40	5	4.9	89	206
5	50	1	23.0	101	192	125.4	52	4	4.3	80	135
6	23	1	22.6	89	139	64.8	61	2	4.2	68	97
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85	220
442	36	1	19.6	71	250	133.2	97	3	4.6	92	57

- **Two hopes**:
  - Accurate prediction
  - Understand important factors

Supervised Learning and Regularization

- **Data**

$$x_i \in \mathcal{X},\ \ y_i \in \mathcal{Y},\ \ i = 1, 2, \ldots, n$$

- Minimize with respect to function $f : \mathcal{X} \to \mathcal{Y}$

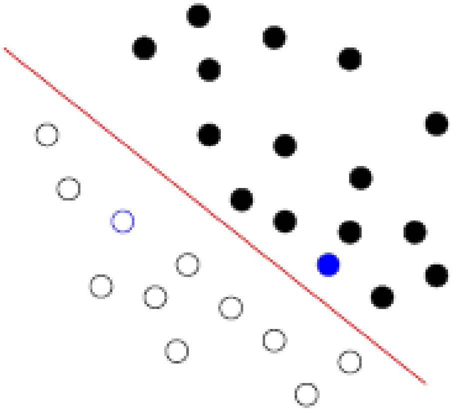
$$\sum_n \ell(f(x_i), y_i) \quad + \quad \frac{\lambda}{2}\|f\|^2$$

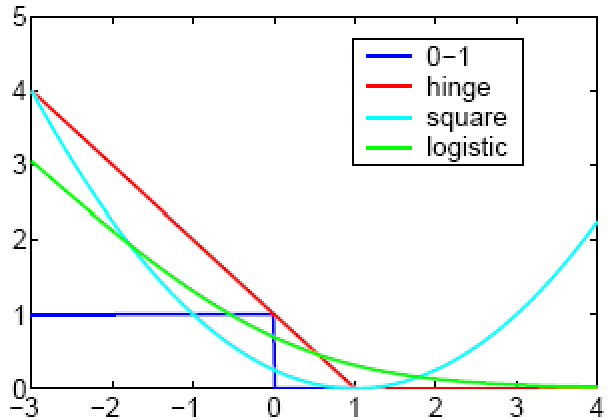
Error on data + Regularization

- Two theoretical/algorithmic issues
  - Loss
  - Function space/norm

Usual Losses

- **Regression**: $y \in \mathbb{R}$, prediction $\hat{y} = f(x)$
  - quadratic cost is $$\ell(y, f) = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - f)^2$$
- **Classification**: $$y \in \{1, -1\},\ \text{prediction } \hat{y} = \text{sign}(f(x))$$
  - loss of the form $\ell(y, f) = \ell(yf)$
  - true loss $\ell(yf) = \delta_{(yf<0)}$
  - useful convex loss





Regularization

- **Main goal**: Avoid over-fitting
- **Two main lines of work**
  - Euclidean and Hilbertian norms (i.e., ℓ_2-norms)
    - Possibility of non linear predictors
    - Non parametric supervised learning and kernel methods
    - Well developed theory and algorithms (see, e.g., Wahba, 1990; Shawe-Taylor and Cristianini, 2004)
  - Sparsity-inducing norms
    - Usually restricted to linear predictors on vectors $f(x) = w^\top x$
    - Main example: ℓ_1-norm

$$\|w\|_1 = \sum_{i=1}^{p} |w_i|$$

    - Perform model selection as well as regularization
    - Theory and algorithms "in the making"

Sparse Linear Estimation with ℓ_1-norm

- The general setting $f : \mathcal{X} \rightarrow \mathcal{Y}$

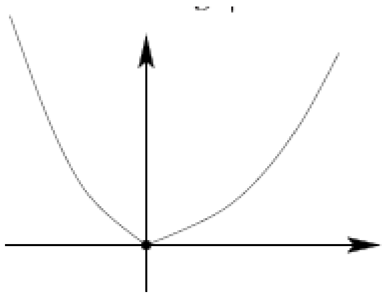
$$\sum_n \ell(f(x_i), y_i) \quad + \quad \Omega(f)$$

- Sparse linear estimation with ℓ_1-norm

$$f(x) = w^\top x$$

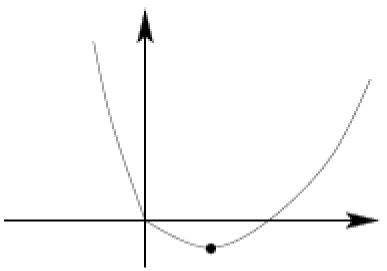
$$\Omega(f) = \|w\|_1 = \sum_{i=1}^p |w_i|$$

Why ℓ_1-norm leads to sparsity?



- **Example 1**: quadratic problem in 1D

$$\min_x \frac{1}{2}x^2 - xy + \lambda|x|$$

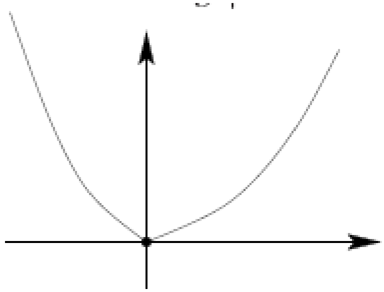


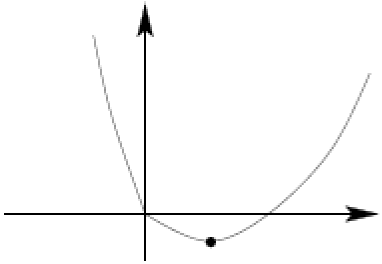
- Piecewise quadratic function with a kink at zero
  - Derivative at 0_+: $g_+ = -y + \lambda$
  - Derivative at 0_- : $g_- = -y - \lambda$
  - $x = 0$ is the solution *iff* $g_+ \geq 0, \; g_- \leq 0 \;$ (i.e.: $|y| \leq \lambda$)
  - $x \geq 0$ is the solution *iff* $g_+ \leq 0 \;$ (i.e.: $y \geq \lambda$) $\; x^* = y - \lambda$
  - $x \leq 0$ is the solution *iff* $g_- \geq 0 \;$ (i.e.: $y \leq -\lambda$) $\; x^* = y + \lambda$
- Solution is: $x^* = \mathrm{sign}(y)(|y| - \lambda)_+$ Soft Thresholding

Why ℓ_1-norm leads to sparsity?

- **Example 1**: quadratic problem in 1D

$$\min_x \frac{1}{2}x^2 - xy + \lambda|x|$$

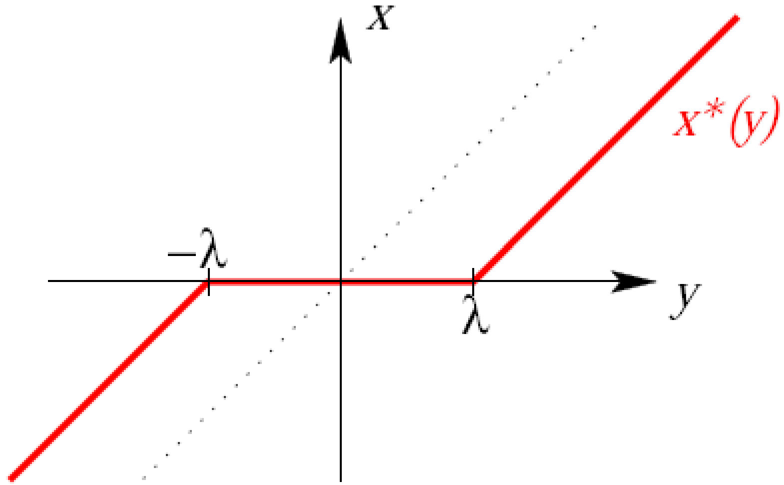




- Piecewise quadratic function with a kink at zero
- Solution is:

Soft Thresholding

$$x^* = \operatorname{sign}(y)(|y| - \lambda)_+$$

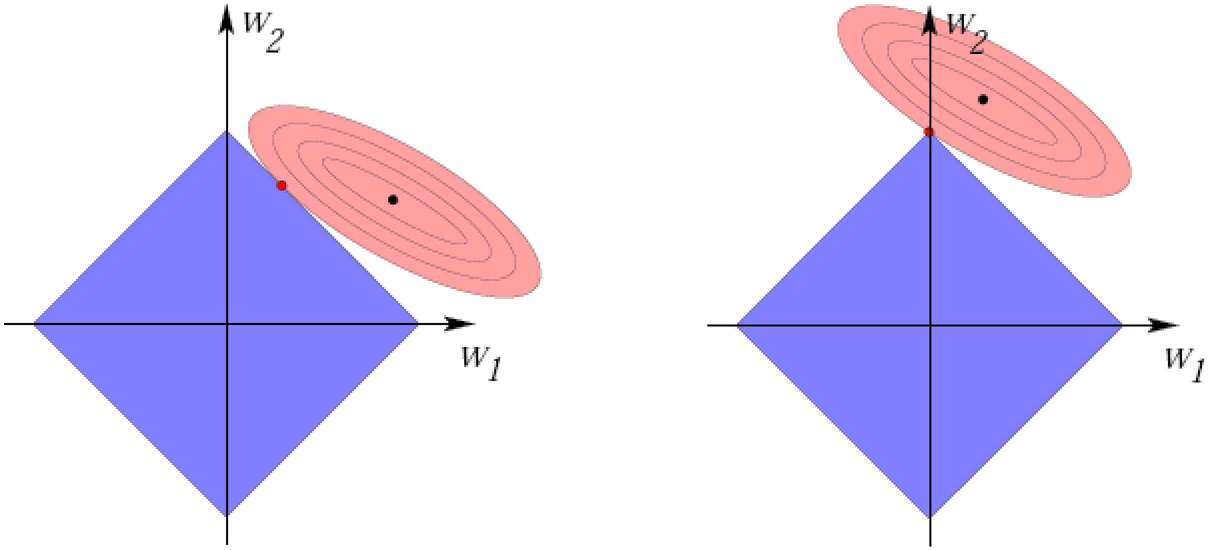


Why ℓ_1-norm leads to sparsity?

- **Example 2**: minimize quadratic function $Q(w)$ subject to

$$\|w\|_1 \leq T$$

- Geometric Interpretation
  - Penalizing is "equivalent" to constraining (HW: proof?)



Sparse Linear Estimation with ℓ_1-norm

- **Data**: covariates $x_i \in \mathbb{R}^p$, response $y_i \in \mathcal{Y}$, $i = 1, 2, \ldots n$
- Minimize over loadings/weights $w \in \mathbb{R}^p$

$$J(w) = \sum_{i=1}^{n} \ell(y_i, w^\top x_i) \quad + \quad \lambda \|w\|_1$$

- Square loss
  - Basis pursuit in signal processing (Chen et al., 1998)
  - Lasso in statistics/machine learning (Tibshirani, 1996)

LASSO

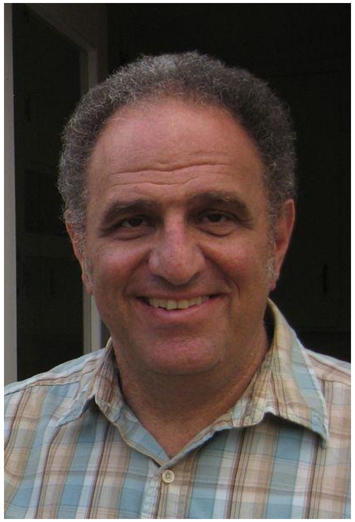
- a loop of rope that is designed to be thrown around a target and tighten when pulled. It is a well-known tool of the American cowboy.

Revisit the Diabetes Study

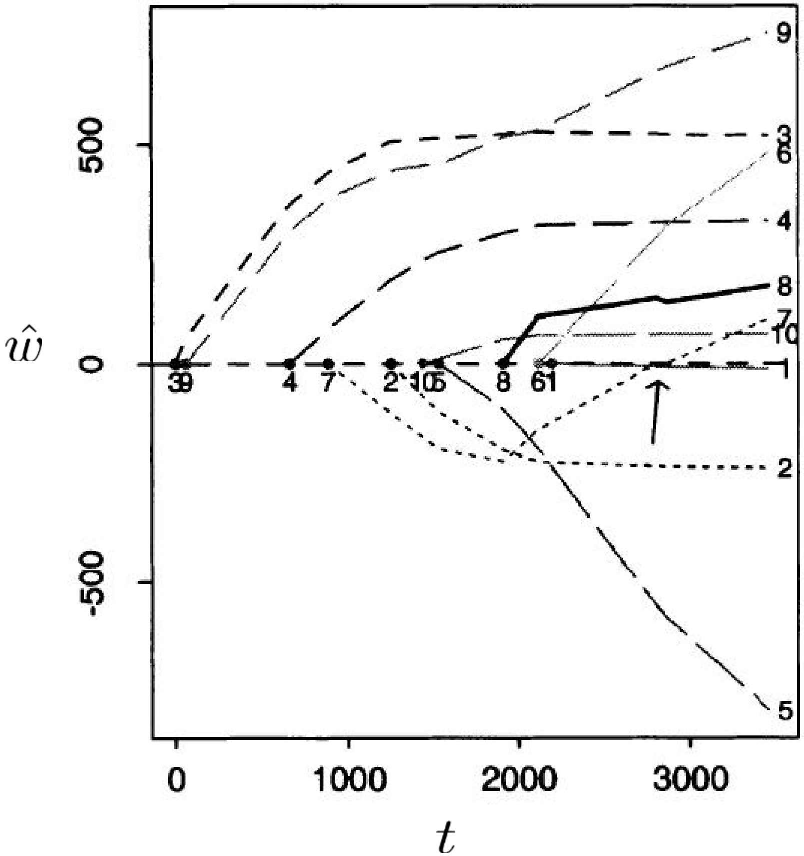
- **Lasso**: least absolute shrinkage and selection operator

$$\min_{w} L(y, w^\top X) = \sum_i (y_i - w^\top x_i)^2$$

$$\text{s.t.: } \|w\|_1 \le t$$



Robert Tibshirani
Dept. Stat., Stanford U.



Nonsmooth convex analysis & optimization

- Analysis
  - optimal conditions
- Optimization
  - algorithms

Optimal conditions for smooth opt. – zero gradient

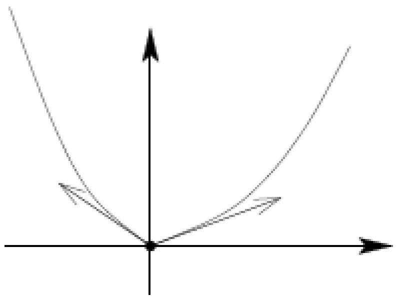
- Example:
$$\min_w J(w) = \sum_{i=1}^n \ell(y_i, w^\top x_i) + \frac{\lambda}{2}\|w\|_2^2$$
  - gradient
$$\nabla J(w) = \sum_i \nabla_w \ell(y_i, w^\top x_i) x_i + \lambda w$$
  - If squared loss
$$\sum_i \ell(y_i, w^\top x_i) = \frac{1}{2}\|y - Xw\|_2^2$$
    - gradient
$$\nabla J(w) = -X^\top (y - Xw) + \lambda w$$
    - solution
$$w = (\lambda I + X^\top X)^{-1} X^\top y$$
- But ℓ_1 -norm is non-differentiable
  - Can't compute the gradient => subgradient (directional derivatives)

Directional Derivatives

- Directional derivative in direction Δ at w :

$$\nabla J(w, \Delta) = \lim_{\epsilon \to 0_+} \frac{J(w + \epsilon\Delta) - J(w)}{\epsilon}$$

  - Rate of change moving through w at the velocity specified by Δ
  - Always exist when *J* is convex and continuous
- **Main idea**: in non-smooth settings, may need to look at all directions



- **Proposition**: *J* is differentiable at w *iff* $\Delta \mapsto \nabla J(w, \Delta)$ is linear

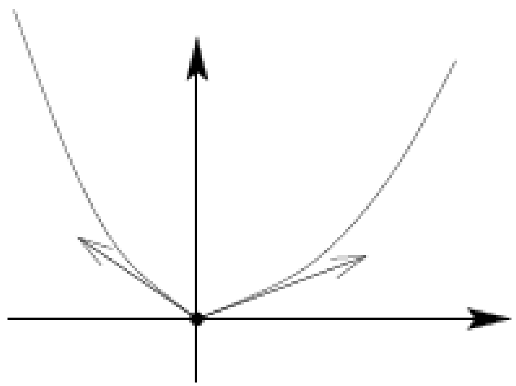
$$\nabla J(w, \Delta) = \nabla J(w)^\top \Delta$$

Optimal conditions for convex functions

- Unconstrained minimization
  - **Proposition**: *w* is optimal *iff*

$$\forall \Delta \in \mathbb{R}^p : \ \nabla J(w, \Delta) \geq 0$$

  - i.e., function value goes up in all directions



- Reduces to zero-gradient for smooth problems?

Directional derivative for ℓ_1-norm

- Function

$$J(w) = \sum_i \ell(y_i, w^\top x_i) + \lambda\|w\|_1 = L(y, Xw) + \lambda\|w\|_1$$

- ℓ_1-norm:

$$\|w + \epsilon\Delta\|_1 - \|w\|_1 = \sum_{j, w_j \neq 0} (|w_j + \epsilon\Delta_j| - |w_j|) + \sum_{j, w_j = 0} |\epsilon\Delta_j|$$

- Thus (separability of optimal conditions)

$$\nabla J(w, \Delta) = \nabla L(w)^\top \Delta + \lambda \sum_{j, w_j \neq 0} \mathrm{sign}(w_j)\Delta_j + \lambda \sum_{j, w_j = 0} |\Delta_j|$$

$$= \sum_{j, w_j \neq 0} (\nabla L(w)_j + \lambda \mathrm{sign}(w_j))\Delta_j + \sum_{j, w_j = 0} (\nabla L(w)_j \Delta_j + \lambda|\Delta_j|)$$

Directional derivative for ℓ_1-norm

- **General loss**: w is optimal *iff* for all $j = 1, 2, \ldots, p$

$$w_j \neq 0 \quad \Rightarrow \nabla L(w)_j + \lambda \text{sign}(w_j) = 0$$

$$w_j = 0 \quad \Rightarrow |\nabla L(w)_j| \leq \lambda$$

- **Squared loss**:

$$L(y, Xw) = \frac{1}{2} \sum_i (y_i - w^\top x_i)^2$$

$$\nabla L(w)_j = -X_j^\top (y - Xw)$$

  - X_j is the j-th column of X

First-order methods for convex opt. – smooth optimization

- **Gradient descent**:

$$w_{t+1} = w_t - \alpha_t \nabla J(w_t)$$

  - with line search: search for a descent α_t
  - with fixed step size, e.g., $\alpha_t = a(t+b)^{-1}$

- Convergence of $f(w_t)$ to $f^*(w) = \min_w f(w)$
  - Depends on the condition number of the optimization number (i.e., correlation within variables)
- **Coordinate descent**:
  - Similar properties

Regularized problems – proximal methods

- Gradient descent as a proximal method

$$w_{t+1} = \arg\max_w L(w_t) + (w - w_t)^\top \nabla L(w_t) + \frac{\mu}{2}\|w - w_t\|_2^2$$
$$= w_t - \frac{1}{\mu}\nabla L(w_t)$$

- Regularized problems of the form $\min_w L(w) + \lambda\Omega(w)$

$$w_{t+1} = \arg\max_w L(w_t) + (w - w_t)^\top \nabla L(w_t) + \lambda\Omega(w) + \frac{\mu}{2}\|w - w_t\|_2^2$$
$$= \mathrm{SoftThreshold}(w_t - \frac{1}{\mu}\nabla L(w_t))$$

- Similar convergence rates as sooth optimization
  - Acceleration methods (Nestrov, 2007; Beck & Teboulle, 2009)

More on Proximal Mapping

- The **proximal mapping** (or proximal operator) of a convex function h is

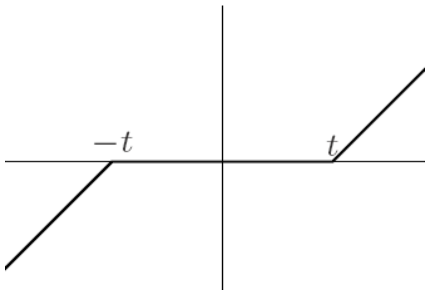
$$\mathrm{prox}_h(x) = \arg\min_{\mu}(h(\mu) + \frac{1}{2}\|\mu - x\|_2^2)$$

- Examples:
  - $h(x) = 0$: $\quad \mathrm{prox}_h(x) = x$
  - $h(x) = I_C(x)$ (indicator function of C): a projection on C

$$\mathrm{prox}_h(x) = P_C(x) = \arg\min_{\mu \in C}\|\mu - x\|_2^2$$

  - $h(x) = t\|x\|_1$: a shrinkage (soft-threshold) operation

$$\mathrm{prox}_h(x)_i = \begin{cases} x_i - t & x_i \geq t \\ 0 & |x_i| \leq t \\ x_i + t & x_i \leq -t \end{cases}$$



More on Proximal Gradient Methods

- **Unconstrained problem** with cost function split in two parts

$$\min_x f(x) = g(x) + h(x)$$

  - g is convex, differentiable
  - h closed, convex, possibly nondifferentiable; prox_h is inexpensive

- **Proximal gradient algorithm**:

$$x^{(k+1)} = \mathrm{prox}_{t_k h}\left(x^{(k)} - t_k \nabla g(x^{(k)})\right)$$

  - t_k is step size, constant or determined by line search

More on Proximal Gradient Methods

$$x^{(k+1)} = \mathrm{prox}_{t_k h}\Big(x^{(k)} - t_k \nabla g(x^{(k)})\Big)$$

- From definition of proximal operator

$$x^{(k+1)} = \arg\min_{\mu} \Big(h(\mu) + \frac{1}{2t_k}\|\mu - x^{(k)} + t_k \nabla g(x^{(k)})\|_2^2\Big)$$

$$= \arg\min_{\mu} \Big(h(\mu) + g(x^{(k)}) + \nabla g(x^{(k)})^\top (\mu - x^{(k)}) + \frac{1}{2t_k}\|\mu - x^{(k)}\|_2^2\Big)$$

  - i.e., minimizes $h(\mu)$ plus a simple quadratic local model of $g(\mu)$ around $x^{(k)}$

Examples

$$\min_x g(x) + h(x)$$

- **Gradient method**: $h(x) = 0$

$$x^{(k+1)} = x^{(k)} - t_k \nabla g(x^{(k)})$$

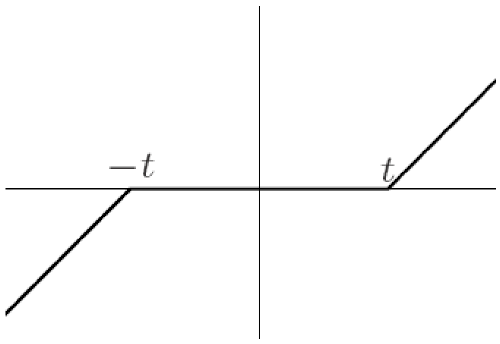
- **Gradient projection method**: $h(x) = I_C(x)$

$$x^{(k+1)} = P_C\left(x^{(k)} - t_k \nabla g(x^{(k)})\right)$$

- **Iterative soft-thresholding**: $h(x) = t\|x\|_1$

$$x^{(k+1)} = \mathrm{prox}_{t_k h}\left(x^{(k)} - t_k \nabla g(x^{(k)})\right)$$

$$\mathrm{prox}_h(x)_i = \begin{cases} x_i - t & x_i \geq t \\ 0 & |x_i| \leq t \\ x_i + t & x_i \leq -t \end{cases}$$



η -Trick for ℓ_1 -norm

- Variational form of the ℓ_1 -norm

$$\|w\|_1 = \min_{\eta \geq 0} \frac{1}{2} \sum_{i=1}^{p} \left(\frac{w_i^2}{\eta_i} + \eta_i \right)$$

- **Alternating minimization**
  - For η , closed-form solution $\eta_i = |w_i|$
  - For *w*, weighted squared ℓ_2 -norm regularized problem
  - **Caveat**: lack of continuity around $(w_i, \eta_i) = (0, 0)$

QP Formulation

- For the special case with square loss

$$\min_{w} \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1$$

  - is equivalent to ($w = w^+ - w^-$)

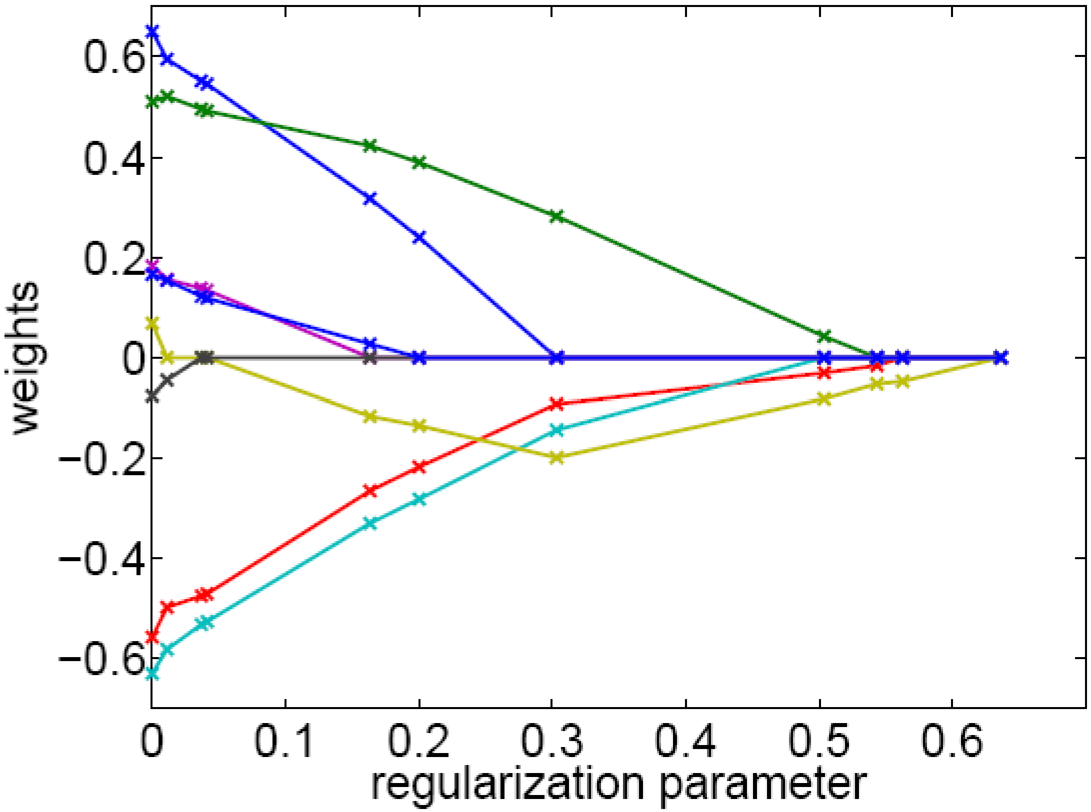
$$\min_{w^+, w^-} \frac{1}{2}\|y - X(w^+ - w^-)\|_2^2 + \lambda \sum_{j=1}^{p} (w_j^+ + w_j^-)$$

$$\text{s.t.: } w^+ \geq 0, \ w^- \geq 0$$

  - generic toolboxes apply, but normally very slow!

Piecewise linear solution paths

- **Lasso**: $$\min_w \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1$$



More on Piecewise Linearity

- The general regularized loss minimization problem

$$\sum_n \ell(f(x_i; w), y_i) \quad + \quad \Omega(w)$$

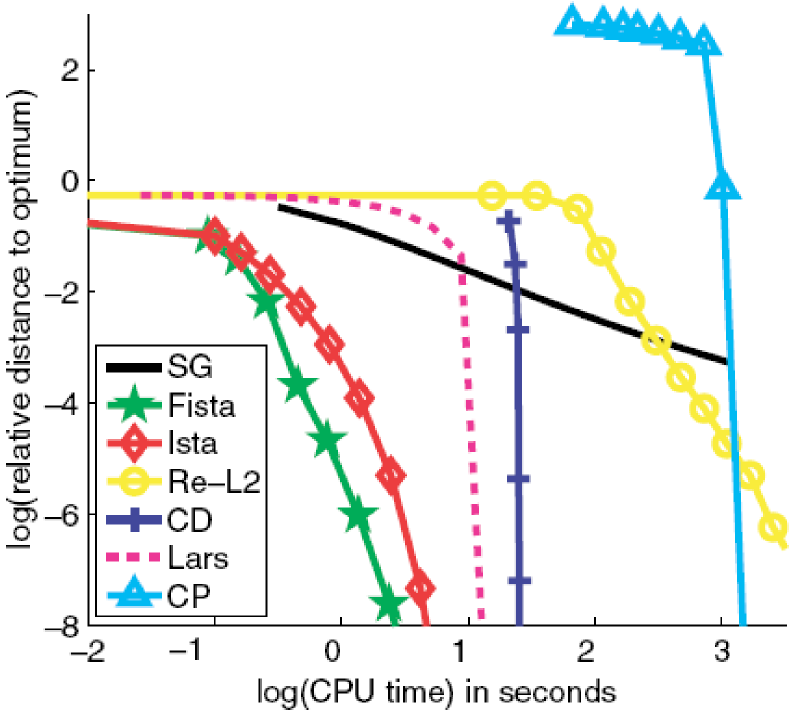
- **Piecewise linearity:**
  - **If** $\exists \lambda_0 = 0 < \lambda_1 < \cdots < \lambda_m = \infty$, and $\gamma_k \in \mathbb{R}^d$
  - **s.t:** $w^*(\lambda) = w^*(\lambda_k) + (\lambda - \lambda_k)\gamma_k$, for $\lambda_k \leq \lambda \leq \lambda_{k+1}$

- **Sufficient conditions** for piecewise linearity
  - ℓ is quadratic or piecewise quadratic as a function of *w*
  - Ω is piecewise linear in *w*

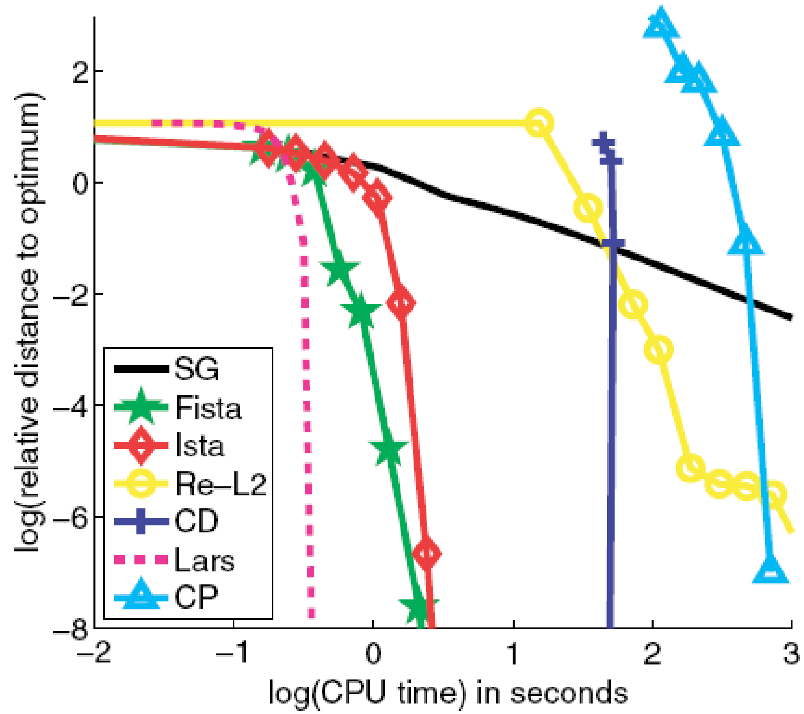
[Rosset and Zhu: Piecewise Linear Regularized Solution Paths, Annals of Stats., 2007]

Comparison on Algorithms for Lasso

$$n = 2000, p = 10,000$$



(a) corr: low, reg: low



(b) corr: low, reg: high

- SG: sub-gradient descent
- Ista: simple proximal methods
- Fista: accelerated version of Ista
- Re-L2: reweighted-least square
- CP: cone programming
- QP: quadratic programming
- Lars: least angle regression
- CD: coordinate descent

Alternative sparse methods
– Greedy methods

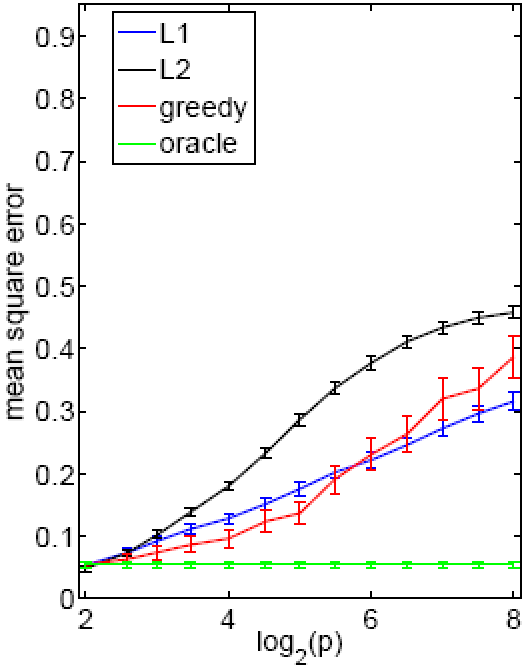
- Forward selection
- Forward-backward selection

- Non-convex method
  - Harder to analyze
  - Simpler to implement
  - Problems of stability

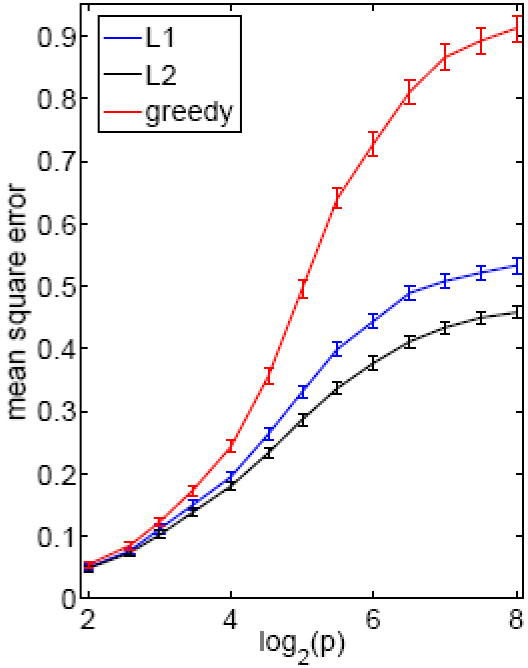
- Positive theoretical results (Zhang, 2009, 2008a)
  - Similar sufficient conditions as for the Lasso

Simulation results

- i.i.d. Gaussian design matrix, $k = 4$, $n = 64$, $p \in [2, 256]$, SNR = 1
- Note stability to non-sparsity and variability



Sparse



Rotated (non sparse)

Summary -- ℓ_1-norm Regularization

- Leads to non-smooth optimization
  - analysis through directional derivatives or subgradients
  - optimization may or may not take advantage of sparsity

- Allows high-dimensional inference

- Interesting problems:
  - Stable variable selection
  - Weaker sufficient conditions (for weaker results)
  - Estimation of regularization parameter (all bounds depend on the unknown noise variance σ2)

Extensions

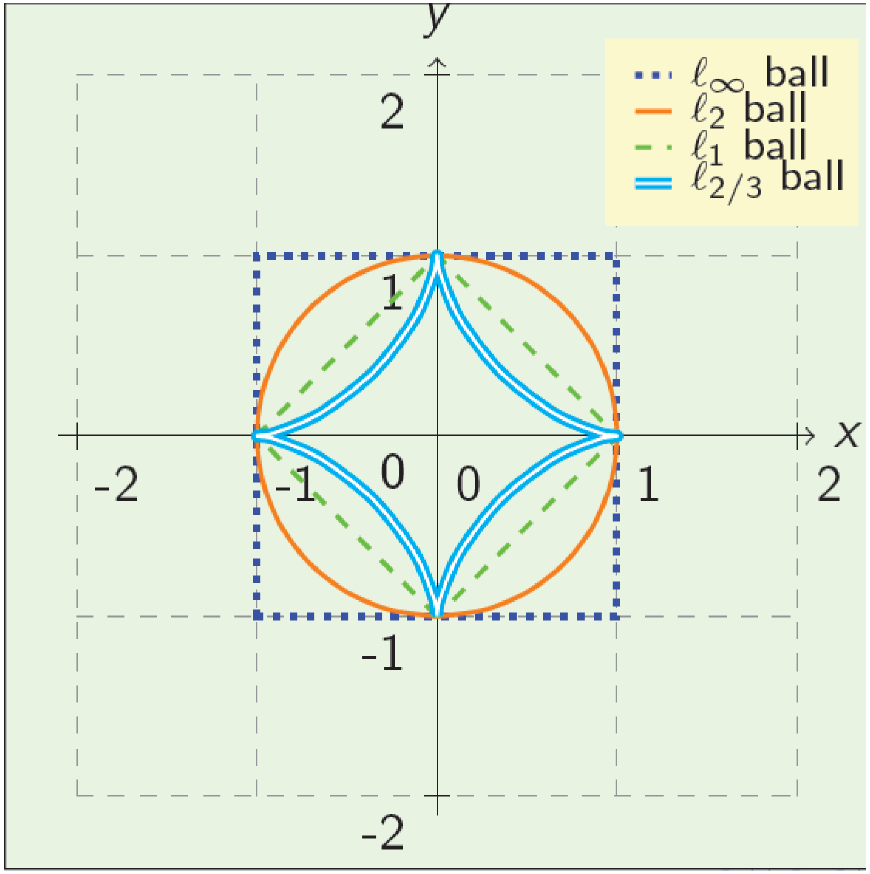
- Sparse methods are not limited to the square loss
  - logistic loss: algorithms (Beck and Teboulle, 2009) and theory (Van De Geer, 2008; Bach, 2009)

- Sparse methods are not limited to supervised learning
  - Learning the structure of Gaussian graphical models (Meinshausen and Buhlmann, 2006; Banerjee et al., 2008)
  - Sparsity on matrices

- Sparse methods are not limited to variable selection in a linear model
  - Kernel learning (Bach et al., 2008)

Extensions

- ℓ_p norm



Regularization with Groups of Variables

- Assume $\{1, 2, \ldots p\}$ is partitioned into m groups $G_1, G_2, \ldots G_m$
- Regularization:

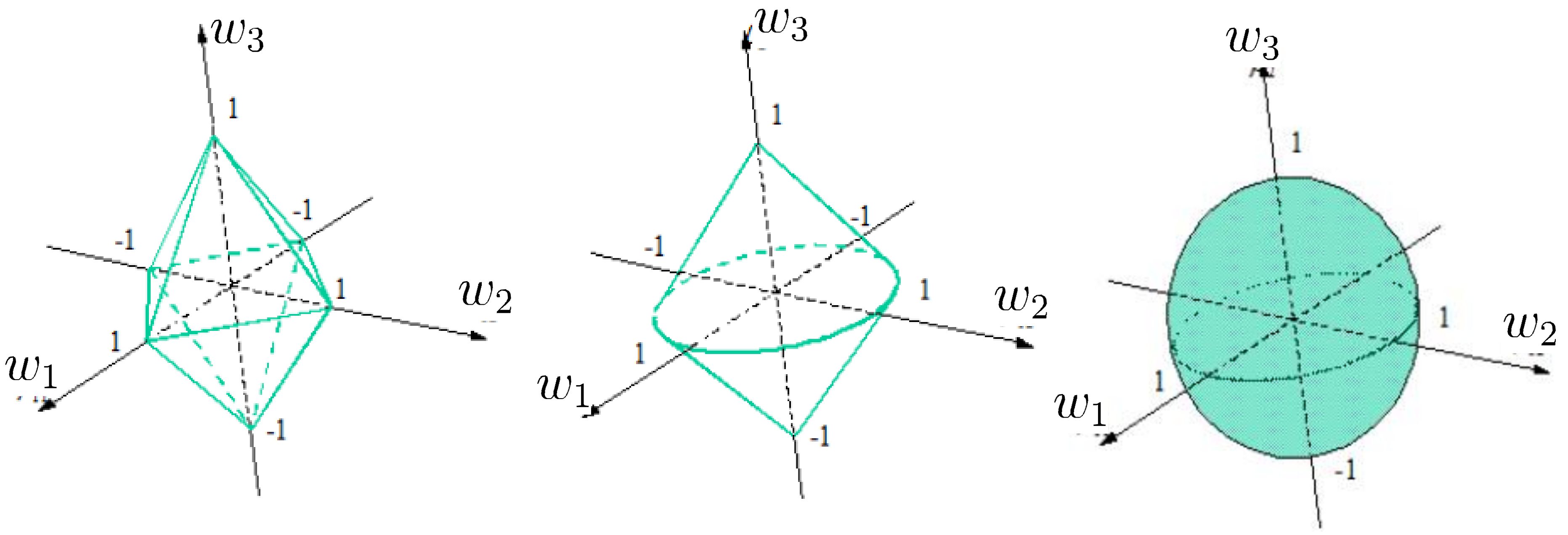
$$\Omega(w) = \sum_{i=1}^{m} \|w_{G_i}\|_2$$

- Induces groupwise sparsity
  - Some groups entirely set to zero
  - No zeros within group

Regularization with Groups of Variables

$$\Omega(w) = \sum_{i=1}^{m} \|w_{G_i}\|_2$$

- E.g.: $\|(w_1, w_2)\|_2 + \|w_3\|_2 \leq 1$



Group Lasso

- **Opt. problem**:

$$\min_{w} \sum_{i} (y_i - w^\top x_i)^2 + \lambda \sum_{i=1}^{m} \sqrt{p_i} \|w_{G_i}\|_2$$

- Optimal condition?
  - **Proposition**: *w* is optimal *iff* $\forall j = 1, 2, \ldots, m$

$$w_{G_j} \neq 0 \quad \Rightarrow -X_{G_j}^\top (y - Xw) + \frac{\lambda \sqrt{p_j} w_{G_j}}{\|w_{G_j}\|_2} = 0$$

$$w_{G_j} = 0 \quad \Rightarrow \|X_{G_j}^\top (y - Xw)\|_2 \leq \lambda \sqrt{p_j}$$

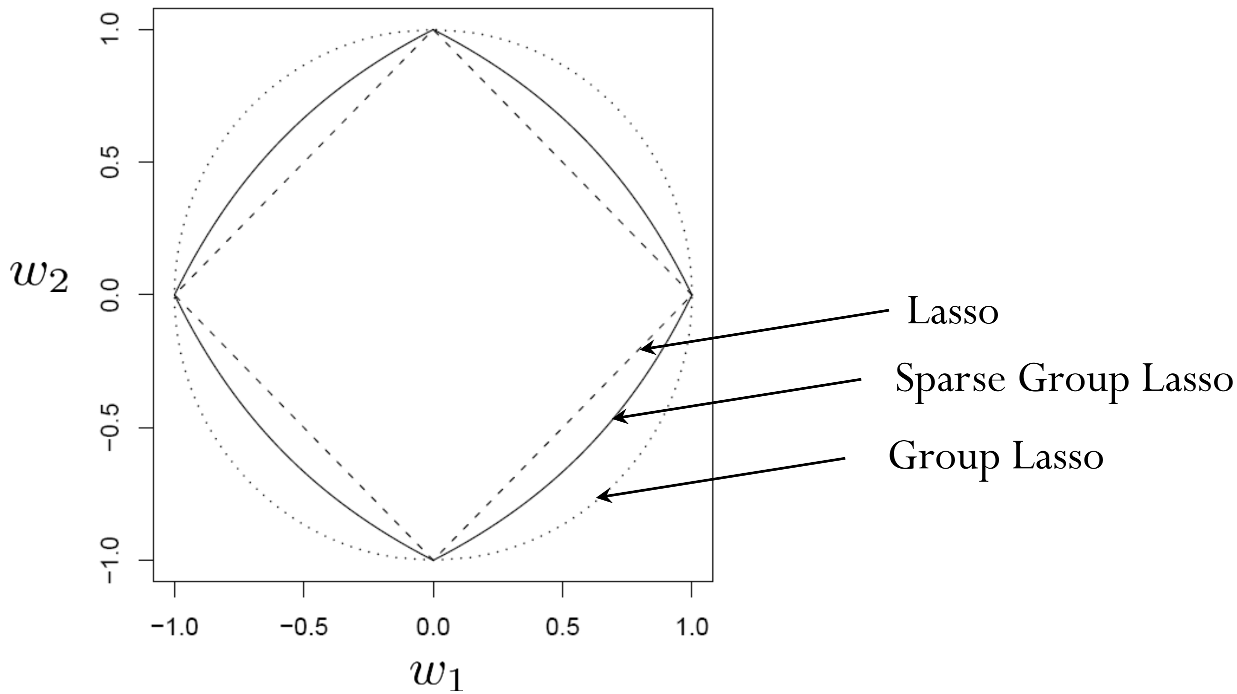
  - p_j is the number of features in group *j*.
- Coordinate descent algorithm can be used to solve it.

Sparse Group Lasso

- **Opt. Problem**:

$$\min_{w} \sum_{i} (y_i - w^\top x_i)^2 + \lambda \sum_{i=1}^{m} \|w_{G_i}\|_2 + \gamma \|w\|_1$$

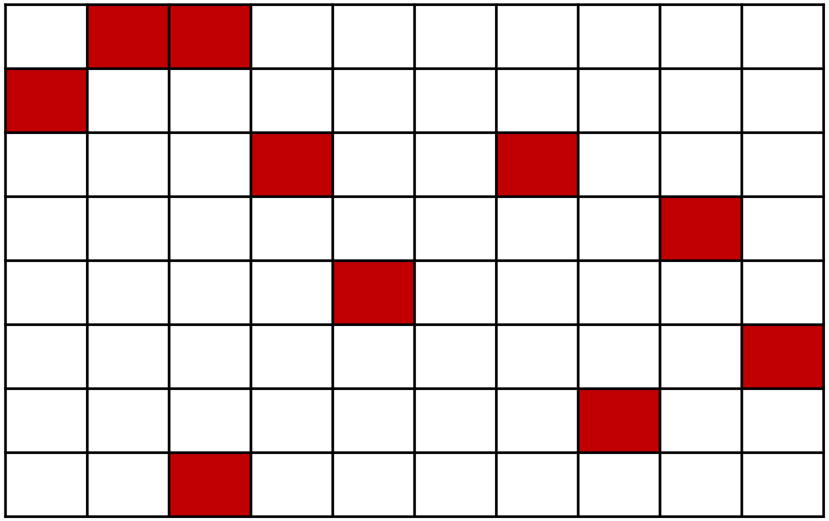
  - the single group case:



Sparsity for Matrices

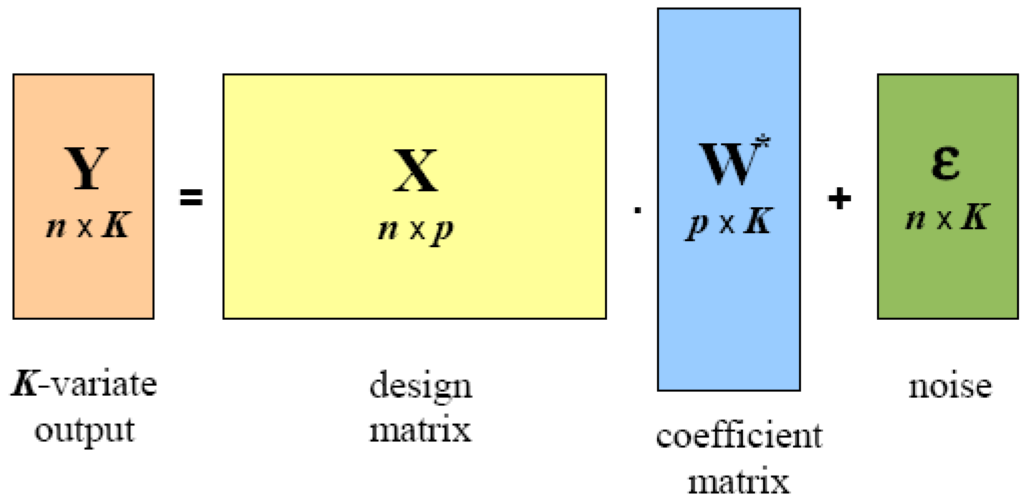
Learning on Matrices
– Collaborative Filtering

- Given N movies and M customers
- Predict the rating of customer i for movie j
- **Training data**: $N \times M$ incomplete matrix that describes the known ratings of some customers for some movies
- **Goal**: complete the matrix



Learning on Matrices
– Multivariate Regression/Classification

- Multivariate linear regression



- Multiclass linear classification

$$\min_{W} \sum_{d=1}^{n} \frac{1}{n} \ell(w_1^\top x_d, \ldots, w_K^\top x_d, y_d)$$

  - where $y_d \in \{0,1\}^K$ and ℓ is the loss, e.g., logistic loss

Learning with Matrices
– Multi-task Learning

- K prediction tasks on the same covariates $x \in \mathbb{R}^p$
  - Each model parameterized by $w_k \in \mathbb{R}^p$
  - Empirical risks:

$$L_k(w_k) = \frac{1}{n}\sum_{i=1}^{n} \ell_k(w_k^\top x_i^k, y_i^k)$$

  - All the parameters form a matrix

$$W = [w_1, \ldots, w_K] = \begin{bmatrix} w_1^1 & \ldots & w_K^1 \\ \vdots & w_k^j & \vdots \\ w_1^p & \ldots & w_K^p \end{bmatrix} = \begin{bmatrix} w^1 \\ \vdots \\ w^p \end{bmatrix}$$

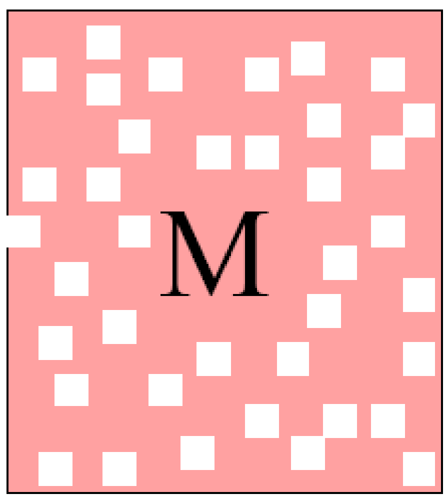
- Many applications:
  - Multi-category classification (one task per class)

Example – Image Denoising

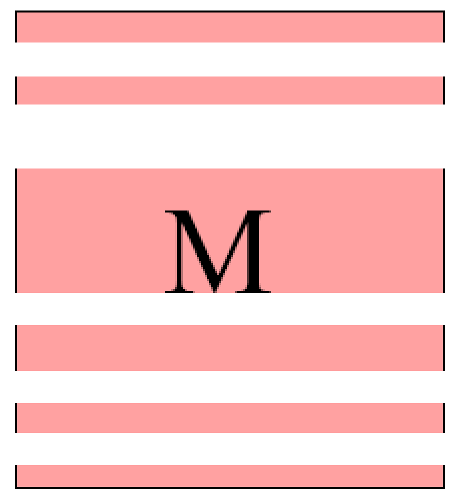
- Simultaneously denoise all patches of a given image
- Example from Mairal et al. (2009)

Two types of sparsity of matrices

- **Type 1 of sparsity:**
  - Directly on the elements



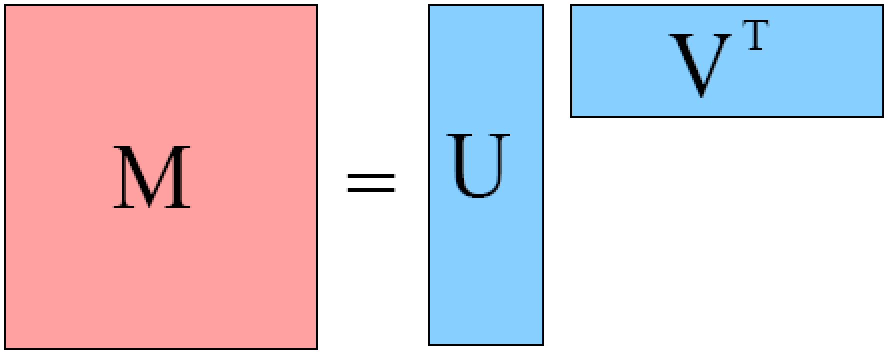
Many elements are zeros



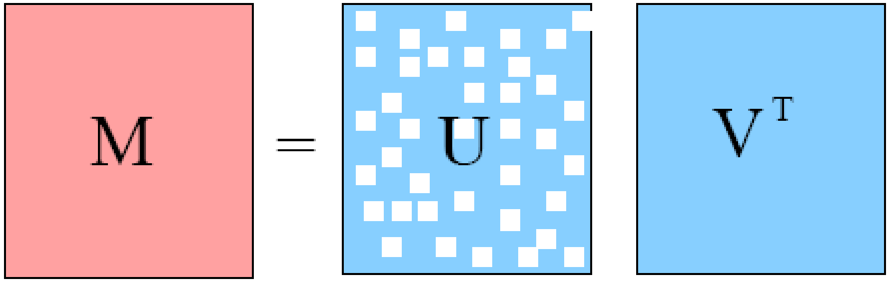
Many rows or columns are zeros

Two types of sparsity of matrices

- **Type 2 of sparsity**:
  - Through a factorization $M = UV^\top$ $U \in \mathbb{R}^{n\times m}$, $V \in \mathbb{R}^{p\times m}$
  - Low-rank sparsity: *m* is small



  - Sparse decomposition: U sparse



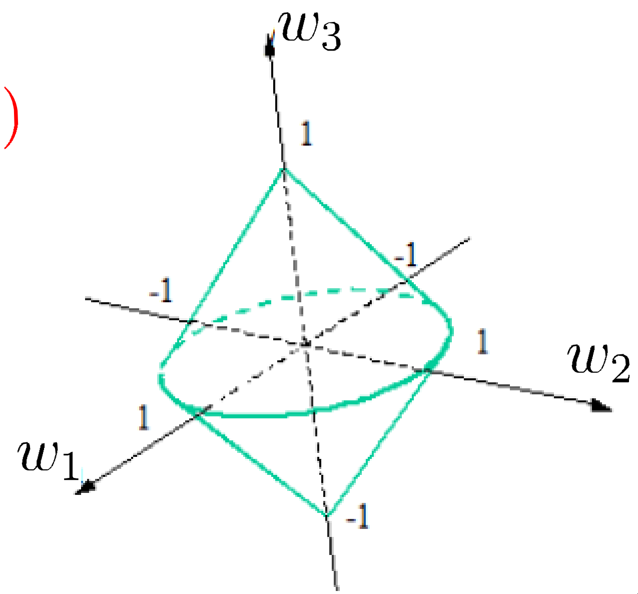
Type 1: Joint Variable Selection in MTL

- Parameters for all the K tasks

$$W = [w_1, \ldots, w_K] = \begin{bmatrix} w_1^1 & \ldots & w_K^1 \\ \vdots & w_k^j & \vdots \\ w_1^p & \ldots & w_K^p \end{bmatrix} = \begin{bmatrix} w^1 \\ \vdots \\ w^p \end{bmatrix}$$

- Select all variables that are relevant to at least one task

$$\min_W \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_k(w_k^\top x_i^k, y_i^k) + \lambda \Omega(W)$$



  - which regularizer?

$$\Omega(W) = \sum_{k=1}^{K} \|w_k\|_2$$

Type 2: Rank constraints and sparsity of the spectrum

- Given a matrix $M \in \mathbb{R}^{n \times p}$
  - Singular value decomposition (SVD): $M = U \mathrm{diag}(\lambda) V^\top$
    where U and V are orthogonal matrices; $\lambda \in \mathbb{R}^m$ are eigenvalues
  - The rank of M is:
    $$\mathrm{rank}(M) = \|\lambda\|_0$$
  - Rank of M is the minimum size m of all factorizations $M = UV^\top$ where $U \in \mathbb{R}^{n \times m}, \; V \in \mathbb{R}^{p \times m}$
- Rank constrained learning

$$\min_{W \in \mathbb{R}^{n \times p}} L(W) \quad \text{s.t.: } \mathrm{rank}(W) \leq m$$

Low-rank via Factorization

- **Reduced-rank multivariate regression**

$$\min_{W \in \mathbb{R}^{n \times p}} \|Y - XW|_F^2 \ \text{ s.t.: } \ \text{rank}(W) \leq m$$

  - Well studied (Anderson, 1951; Izenman, 1975; Reinsel and Velu, 1998)
  - Is solved directly using the SVD (by OLS + SVD + projection)

- **General formulation**

$$\min_{U \in \mathbb{R}^{n \times m}, V \in \mathbb{R}^{p \times m}} L(UV^\top)$$

  - Still non-convex but convex w.r.t. U and V separately
  - Optimization by alternating procedures

Trace-norm Relaxation

- With SVD $M = U\mathrm{diag}(\lambda)V^\top$ $\mathrm{rank}(M) = \|\lambda\|_0$
  - 0-norm can be relaxed by 1-norm $\|\lambda\|_1$
  - This is the trace norm, denoted by $\|M\|_{\mathrm{tr}} = \|\lambda\|_1$

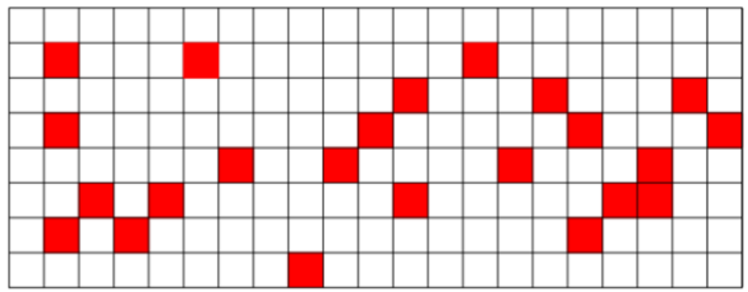
- **Trace-norm regularized learning**

$$\min_{W \in \mathbb{R}^{n \times p}} L(W) + \lambda \|W\|_{\mathrm{tr}}$$

  - Convex problem
  - Can be solved with: proximal methods; Iterative re-weighted Least-squares; etc

Trace-norm and Collaborative Filtering

- **CF as matrix completion** (users as rows; items as columns)

$Y =$ 

- Find a low-rank matrix to reconstruct noisy observations

$$\min_{X \in \mathbb{R}^{n \times p}} \sum_{(i,j) \in S} (X_{ij} - Y_{ij})^2 + \lambda \|X\|_{\mathrm{tr}}$$

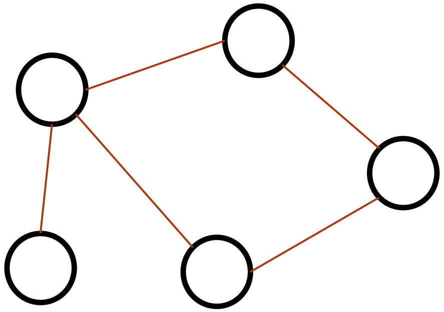
  - Semi-definite program (Fazel et al., 2001)
  - Max-margin approaches to CF (Srebro et al., 2005)
  - High-dimensional inference fro noisy matrix completion (Srebro et al., 2005; Candes & Plan, 2009)
  - May recover entire matrix from slightly more entries than the minimum of the two dimensions

Graphical Lasso

- aka: **sparse inverse covariance estimation**
- Gaussian graphical models
  - A set of random variables $X_1, \cdots, X_N$
  - The joint distribution is multivariate Gaussian

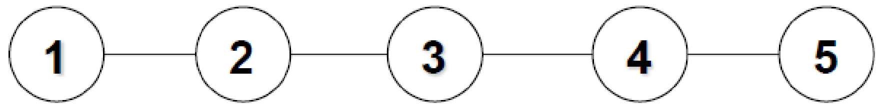
$$p(\mathbf{x}) = \mathcal{N}(\mu, \Sigma)$$

- **Proposition (sparse structure)**:
  - If the ij-th element of Σ^{-1} is zero, then X_i and X_j are conditionally independent, i.e., no direct edge



Gaussian Random Fields

- An example



$$\Sigma^{-1} = \begin{pmatrix} 1 & 6 & 0 & 0 & 0 \\ 6 & 2 & 7 & 0 & 0 \\ 0 & 7 & 3 & 8 & 0 \\ 0 & 0 & 8 & 4 & 9 \\ 0 & 0 & 0 & 9 & 5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 0.10 & 0.15 & -0.13 & -0.08 & 0.15 \\ 0.15 & -0.03 & 0.02 & 0.01 & -0.03 \\ -0.13 & 0.02 & 0.10 & 0.07 & -0.12 \\ -0.08 & 0.01 & 0.07 & -0.04 & 0.07 \\ 0.15 & -0.03 & -0.12 & 0.07 & 0.08 \end{pmatrix}$$

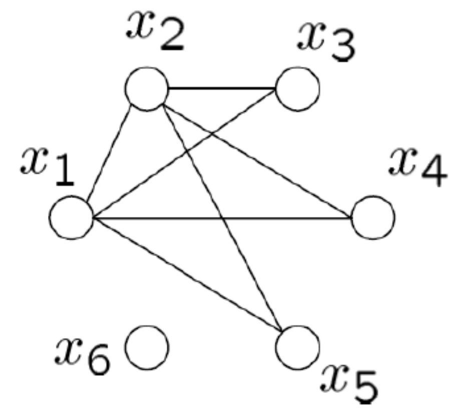
$$\Sigma_{15}^{-1} = 0 \Leftrightarrow X_1 \perp X_5 \,\big|\, X_{nbrs(1) \text{ or } nbrs(5)}$$

$$\not\Rightarrow$$

$$X_1 \perp X_5 \Leftrightarrow \Sigma_{15} = 0$$

Another example

$$Q = \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & * & * & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & 0 & * & 0 & 0 \\ * & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$



- How to estimate this MRF?
- What if p >> n?
  - MLE doesn't exist in general!
  - What about only learning a "sparse" graphical model?
    - This is possible when s=o(n)
    - Very often it is the structure of the GM that is more interesting …

Graphical Lasso

- aka: **sparse inverse covariance estimation**

- Sparse learning problem:
  - Let $\Theta = \Sigma^{-1}$

$$\min_{\Theta} -\log p(\mathbf{X}|\Theta) + \lambda\|\Theta\|_1$$

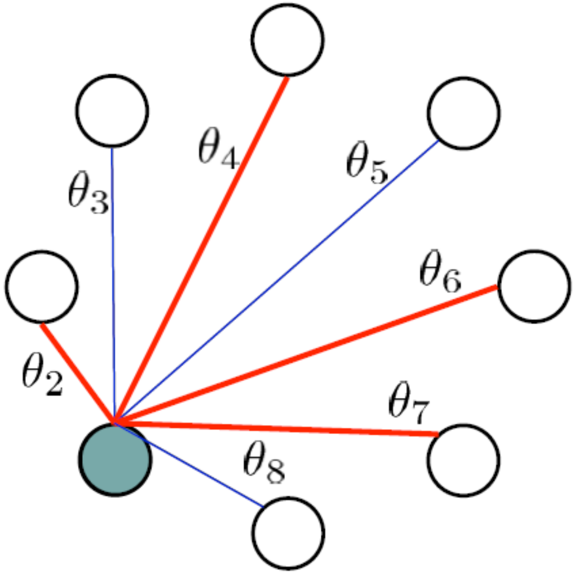
- Various algorithms:
  - Banerjee et al. (2007): block coordinate descent
  - Friedman et al. (2008): graphical lasso
  - …

Sparse Learning of General Graphs

- Local methods
  - Sparse-norm regularized logistic regression + aggregation
  - See (Wainwright et al., 2006)
- Global methods
  - Sparse-norm regularized MLE
  - See (Lee et al., 2006; Zhu et al., 2010; etc.)

Wainwright et al: High-Dimensional Graphical Model Selection Using ℓ1-Regularized Logistic Regression
Lee et al: Efficient structure learning of Markov networks using ℓ1-regularization
Zhu et al: Grafting-Light: Fast, Incremental Feature Selection and Structure Learning of MRFs

Graphical Regression

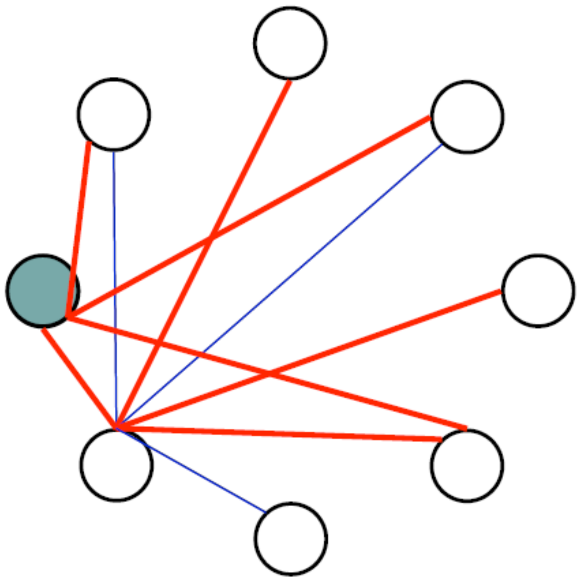


Neighborhood selection

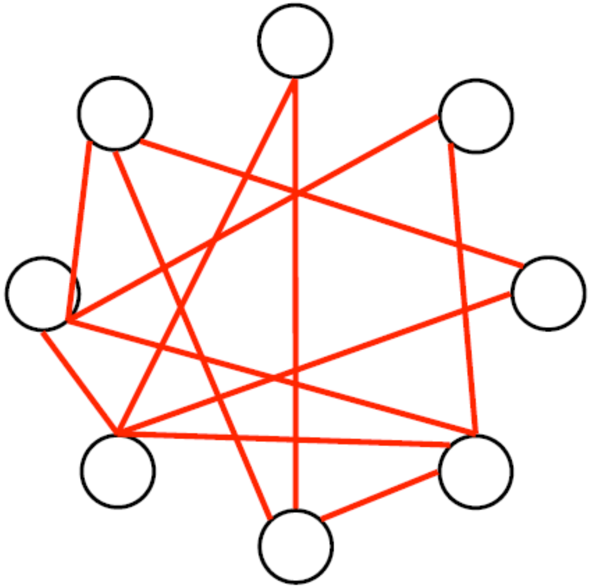
Lasso:

$$\hat{\theta} = \arg\min_{\theta} \sum_{t=1}^{T} l(\theta) + \lambda_1 \| \theta \|_1$$

Graphical Regression



Graphical Regression



Consistency

- Theorem: for the graphical regression algorithm, under certain verifiable conditions (omitted here for simplicity):

$$\mathbb{P}\left[\hat{G}(\lambda_n) \neq G\right] = \mathcal{O}\left(\exp\left(-Cn^{\epsilon}\right)\right) \to 0$$

Dictionary Learning

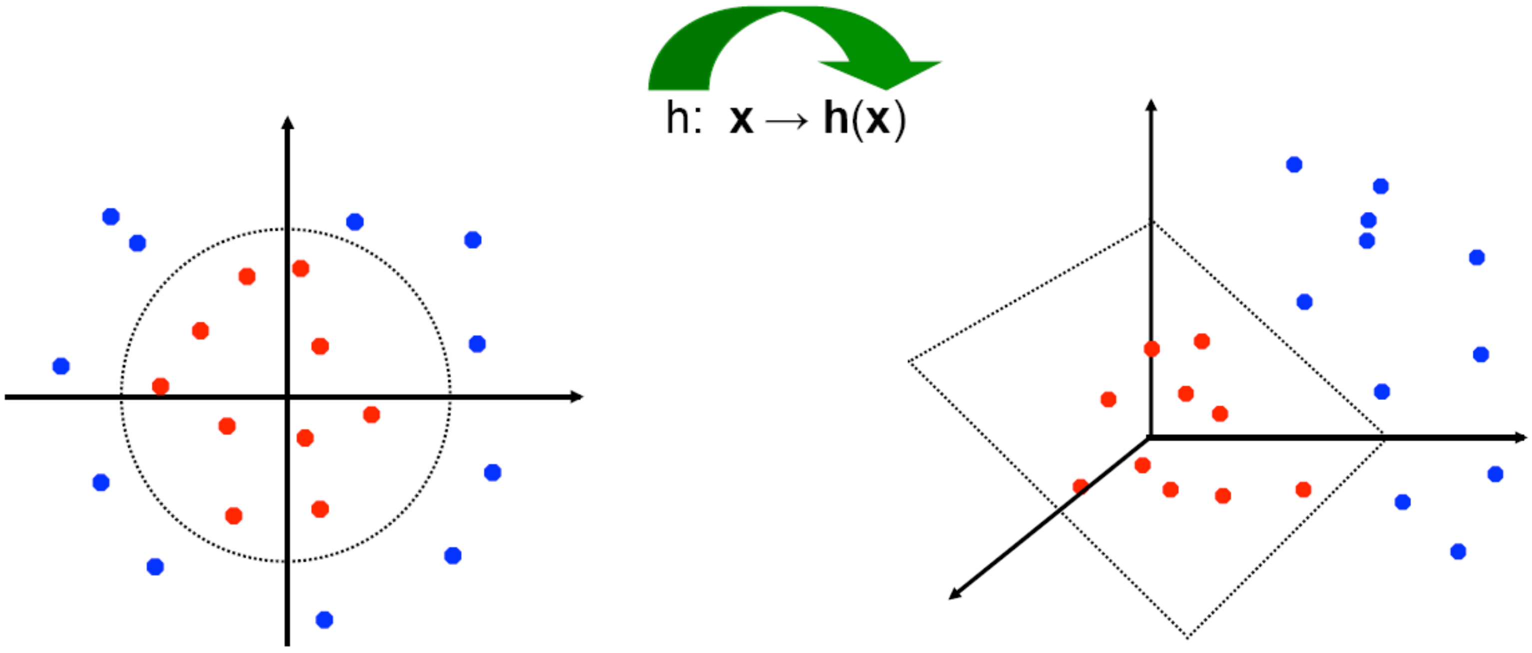
Learning with pre-defined basis functions -- generalized linear models

- A mapping function

$$\phi : \ \mathcal{X} \rightarrow \mathbb{R}^N$$

- Doing linear regression in the mapped space

$$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$$



Fixed Basis Functions

- Given a set of basis functions $\{\phi_h(\mathbf{x})\}_{h=1}^{H}$

$$\phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \cdots \phi_H(\mathbf{x})]^\top$$

  - E.g. 1:

$$\phi_h(\mathbf{x}) = \exp\left(- \frac{\|\mathbf{x} - c_h\|_2^2}{2r^2} \right)$$

  - E.g. 2:

$$\phi_h(\mathbf{x}) = x_i^p x_j^q$$

$$\boxed{f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}}$$

Dictionary Learning

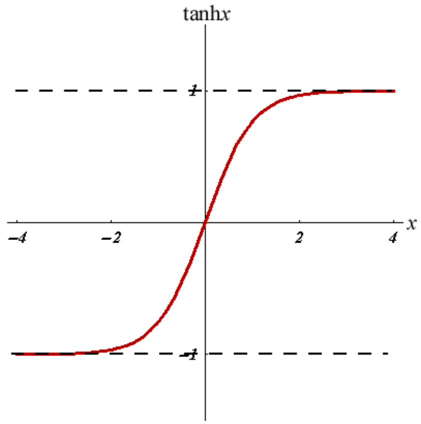
- **Goal**:
  - learn the basis functions from data

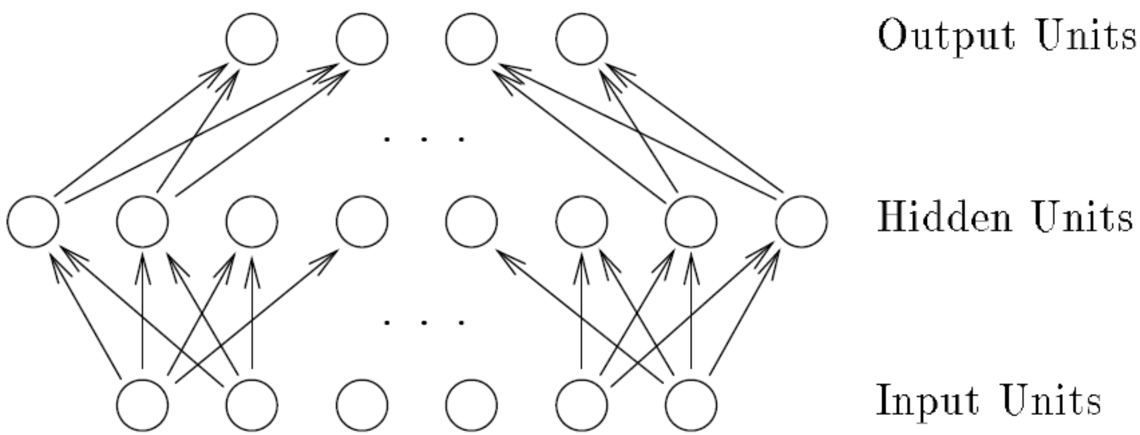
Parametric Basis Functions

- Neural networks to learn a parameterized mapping function
- E.g., a two-layer feedforward neural networks

$$\phi_h(\mathbf{x}) = \tanh\left(\sum_{i=1}^{I} w_{hi}^{(1)} x_i + w_{h0}^{(1)}\right)$$

$$f(\mathbf{x}; \mathbf{w}) = \sum_{h=1}^{H} w_h^{(2)} \phi_h(\mathbf{x}) + w_0^{(2)}$$





[Figure by Neal]

PCA: minimum error formulation

- A set of complete **orthonormal basis**

$$\{\mu_i\},\ i = 1, \ldots, D$$

$$\mu_i^\top \mu_j = \delta_{ij}$$

- We consider a low-dimensional approximation

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni}\mu_i + \sum_{i=M+1}^{D} b_i \mu_i$$

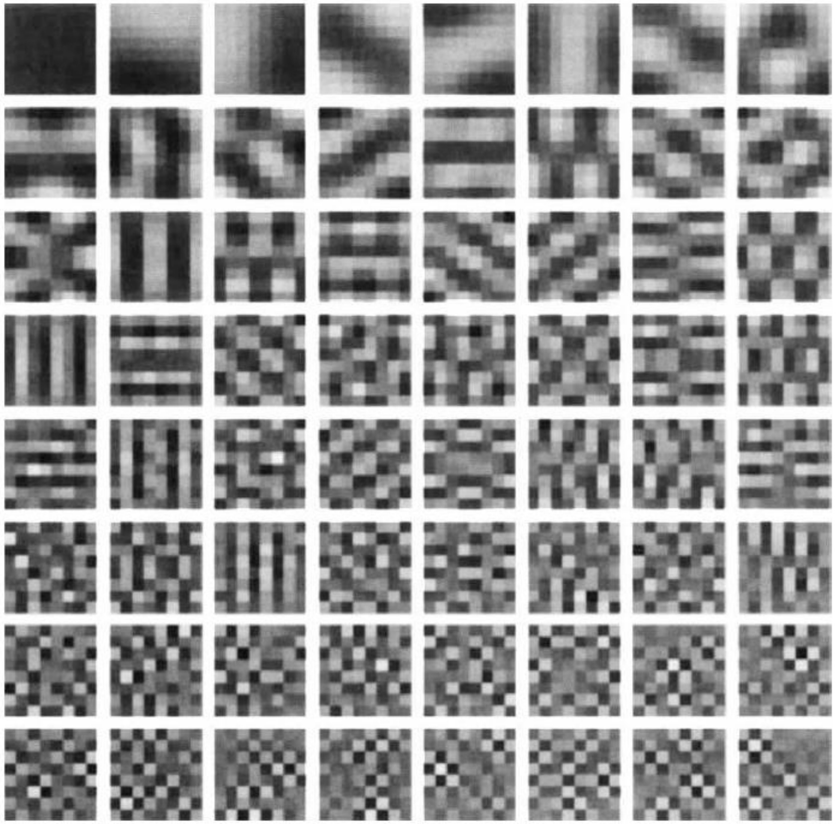
- The best approximation is to minimize the error

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$$

Issues with PCA

- Principal components calculated on 8x8 image patches

  - PCA capture linear pairwise statistics
  - Suitable for Gaussian distributed data



  - Not localized
  - Not resemble cortical receptive fields
  - Not suitable for images with high order statistics

[Olshausen & Field, Nature 1996]

Sparse Coding

- **Basic assumption 1**: a linear superposition model

$$I(x,y) = \sum_i \alpha_i \phi_i(x,y)$$

an image

basis function

- **Basic assumption 2**: nature images have 'sparse structure' (similar as minimum-entropy code)

$$\alpha = \begin{pmatrix} 0.4 \\ 0 \\ 0 \\ 0.1 \\ 0.2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

[Olshausen & Field, Nature 1996]

Sparse Coding

- Search for a sparse code is an optimization problem:

$$\min_{\alpha,\phi} \sum_{x,y} \left[I(x,y) - \sum_i \alpha_i \phi_i(x,y) \right]^2 + \psi(\alpha)$$

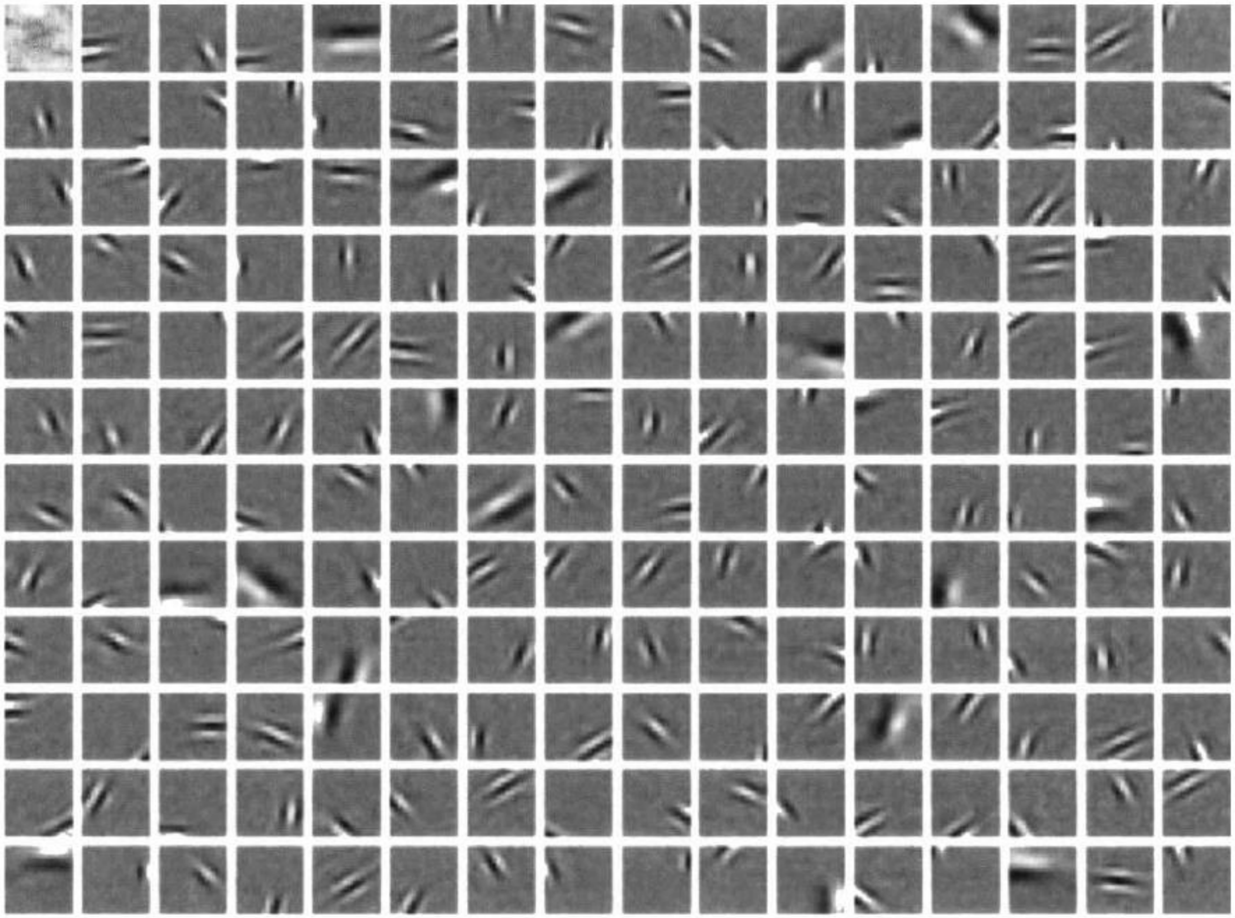
  - the L1-norm is a common choice

- Solve the problems – alternating minimization
  - For each image, solve for α as a sparse learning problem
  - Update dictionary using gradient descent

[Olshausen & Field, Nature 1996]

Sparse Coding

- Basis learned on 16 x 16 natural scene image patches



- Localized
- Oriented
- Selective to spatial scales

[Olshausen & Field, Nature 1996]

Nonnegative Matrix Factorization

- Matrix factorization

$$\min_{U \in \mathbb{R}^{n \times m}, V \in \mathbb{R}^{p \times m}} L(UV^\top; X)$$

  - Example losses:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{p} ((UV^\top)_{ij} - X_{ij})^2$$

$$L = \sum_{i=1}^{n} \sum_{j=1}^{p} (X_{ij} \log(UV^\top)_{ij} - (UV^\top)_{ij})$$

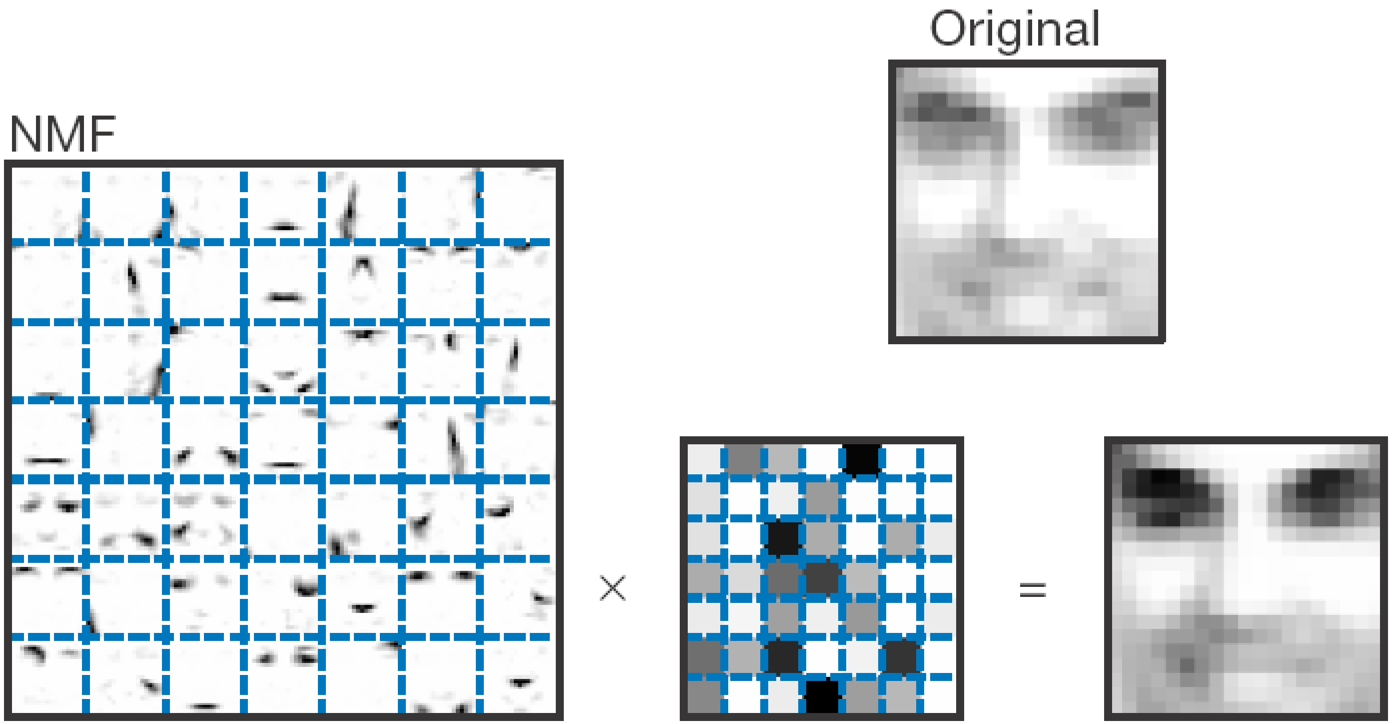
- Non-negativity loss:

$$U \geq 0; \;\; V \geq 0$$

[Lee & Seung, Nature 1999]

Nonnegative Matrix Factorization

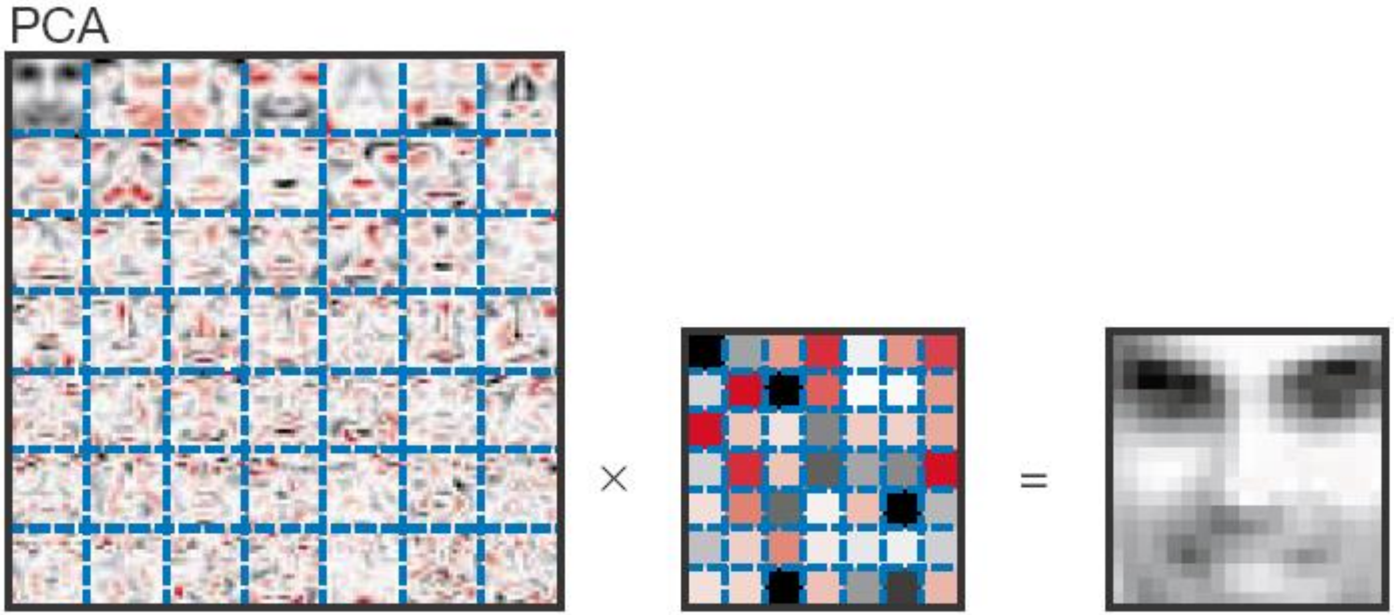
- Sparse basis and sparse coefficients for images:



[Lee & Seung, Nature 1999]

Nonnegative Matrix Factorization

- Eigenfaces and non-sparse coefficients by PCA
  - Positive and negative combinations



[Lee & Seung, Nature 1999]

Nonnegative Matrix Factorization

- NMF for text documents with bag-of-word counts

$$X \approx UV$$

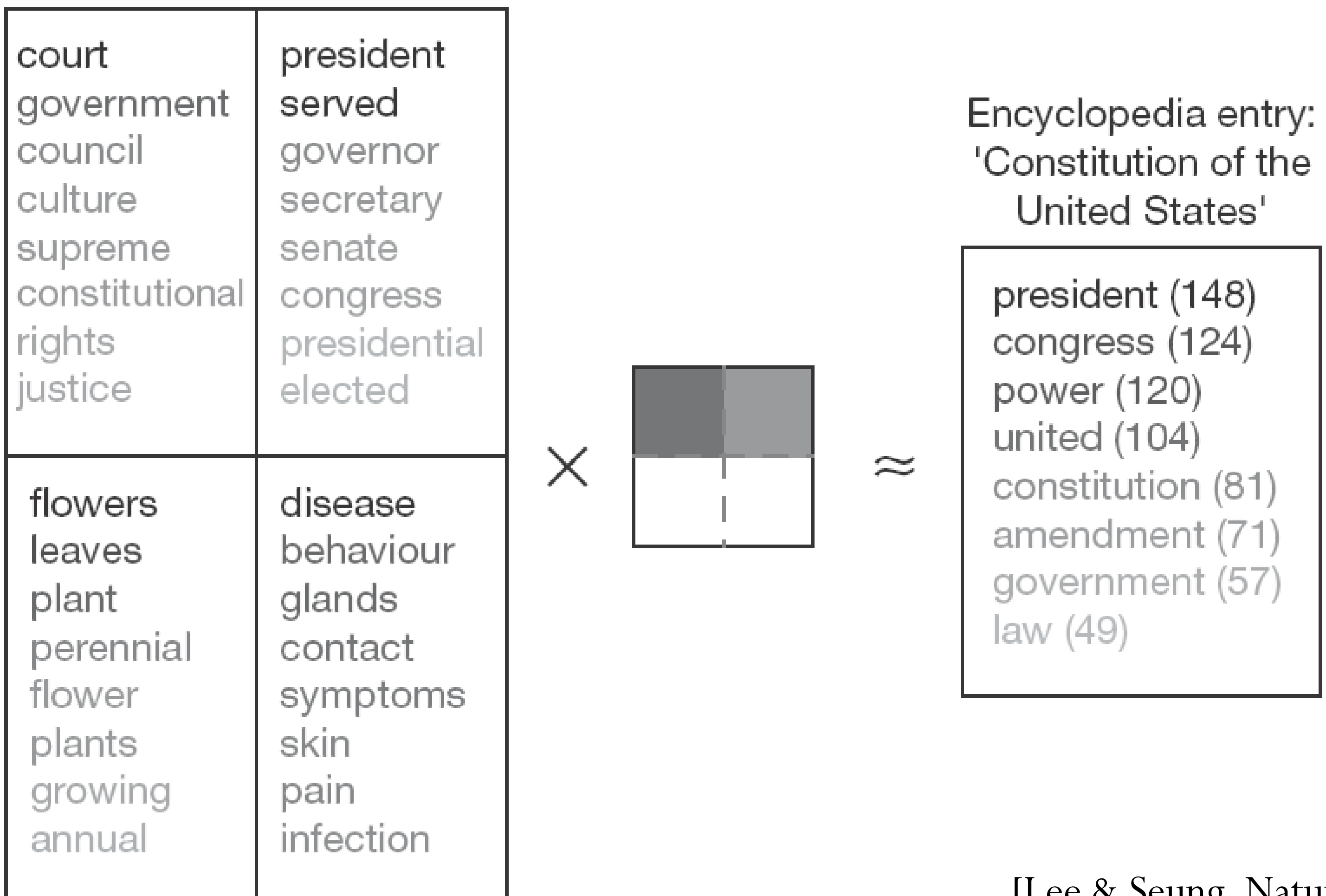
$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \cdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{pmatrix}_{N \times D}$$

$$X_d = \begin{pmatrix} X_{1d} \\ X_{2d} \\ \vdots \\ X_{Nd} \end{pmatrix}_{N \times 1} \approx \begin{pmatrix} | & | & \cdots & | \\ U_{.1} & U_{.2} & \cdots & U_{.K} \\ | & | & \cdots & | \end{pmatrix}_{N \times K} \times \begin{pmatrix} V_{1d} \\ V_{2d} \\ \vdots \\ V_{Kd} \end{pmatrix}_{K \times 1}$$

- The same coefficient vector to reconstruct all word counts in a document

Nonnegative Matrix Factorization

- NMF for text documents with bag-of-word counts



[Lee & Seung, Nature 1999]

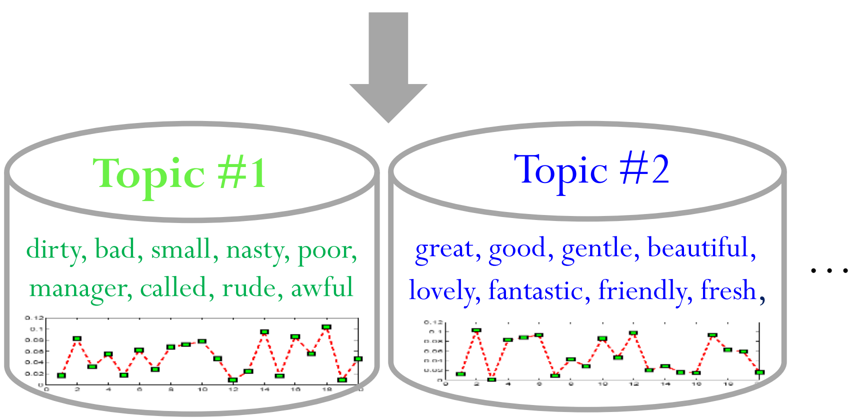
Topic Modeling – projection view

Dictionary Learning

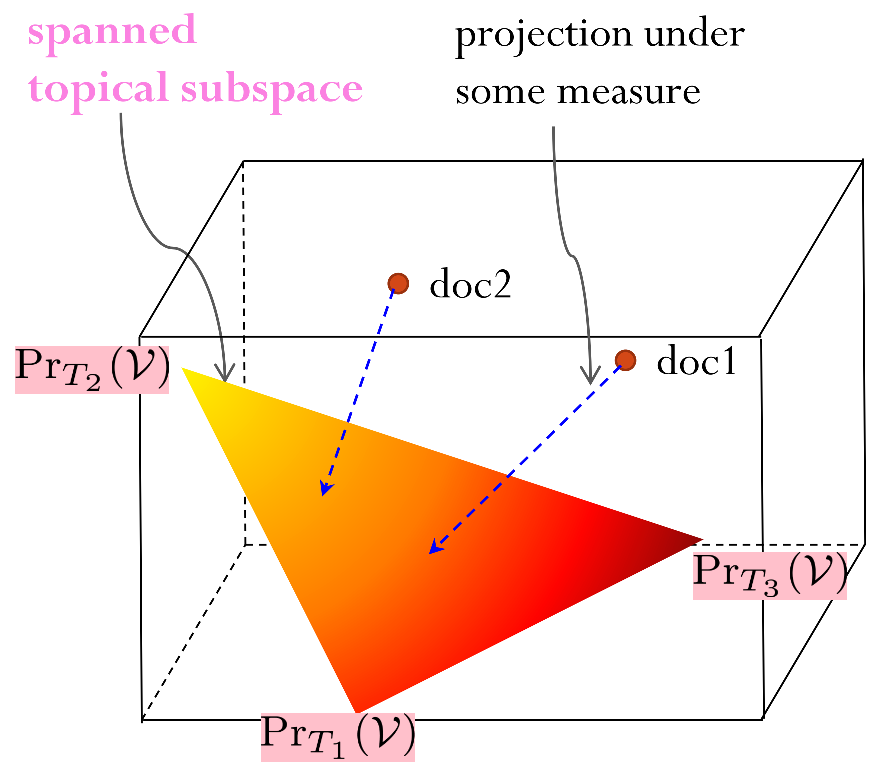
doc1: The room was **dirty** and smelled **awful**. The wallpaper was **dirty** and **nasty**. The hallway smelled **awful** as did the water. The price was **awful** 100.00 to stay in such a **dirty** room. Just **awful**. I would not recommend this hotel to an **enemy**.

doc2: **Fantastic** Hotel in **beautiful** Krabi. We felt pampered by the **gentle** and **caring** staff. The turndown service each evening maintains the very **high** standard. A **blissful** sleep in the most **comfortable** beds. The food is of a extremely **high** quality with **fantastic** **fresh** and **vibrant** tastes.

⋮



Topical Projection

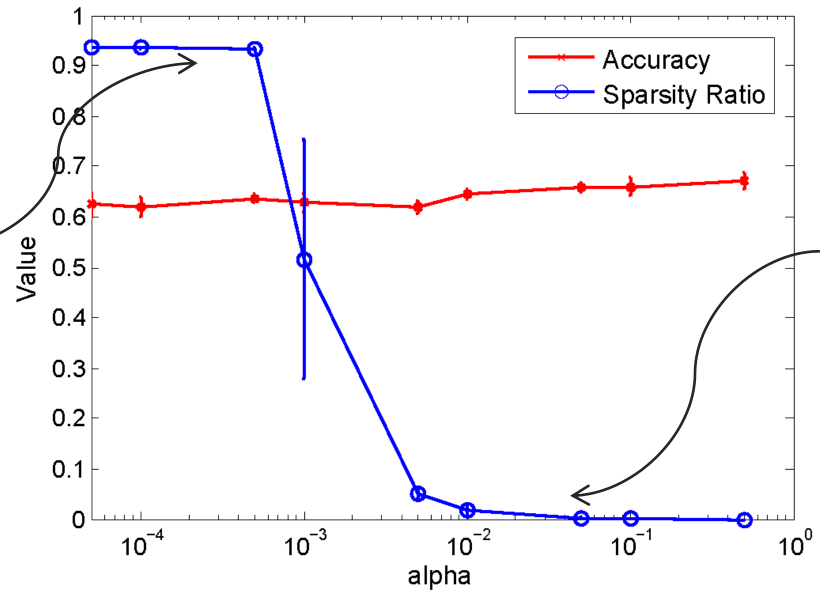


- **Probabilistic topic models**:
  - Topical subspace is a simplex
  - Projection under KL-divergence

Probabilistic Topic Models – restrictions

- Ineffective in controlling posterior sparsity by using priors, e.g., Dirichlet prior in LDA (Zhu & Xing, UAI 2011):

weak smoothing:
sparse but worse predict accuracy

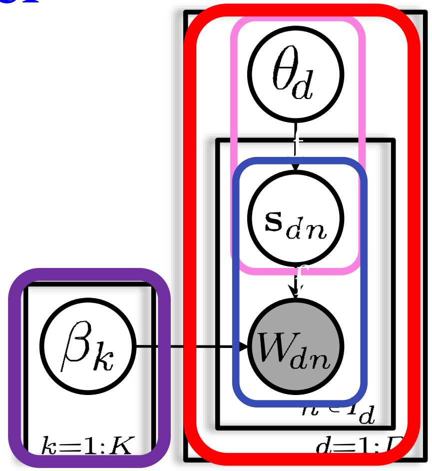


strong smoothing:
improved prediction, but non-sparse

- Restricted to MLE when considering supervised side information;
- Hard in inference due to a normalized likelihood model when considering discrete side information (e.g., category labels or features)

Sparse Topical Coding (STC)

A Non-probabilistic Topic Model



- Topical bases:

$$\beta_k \in \mathcal{P}_N \qquad \beta = \begin{pmatrix} | & | & \cdots & | \\ \beta_{.1} & \beta_{.2} & \cdots & \beta_{.N} \\ | & | & \cdots & | \end{pmatrix}_{K \times N}$$

- Hierarchical coding:
  - **word code $\mathbf{s}$** – encode word counts under a loss:

$$\ell(w_n, \mathbf{s}_n, \beta) = \log p(w_n | \mathbf{s}_n, \beta)$$

where $\mathbb{E}_{p(w_n|\mathbf{s}_n,\beta)}[T(w_n)] = \mathbf{s}_n^\top \beta_{.n}$ we use *Poisson* distribution

  - **document code θ** – an aggregation of word codes

$$\sum_{n \in I} \|\mathbf{s}_n - \theta\|_2^2$$

- Nonnegative hierarchical sparse coding (with dictionary learning)

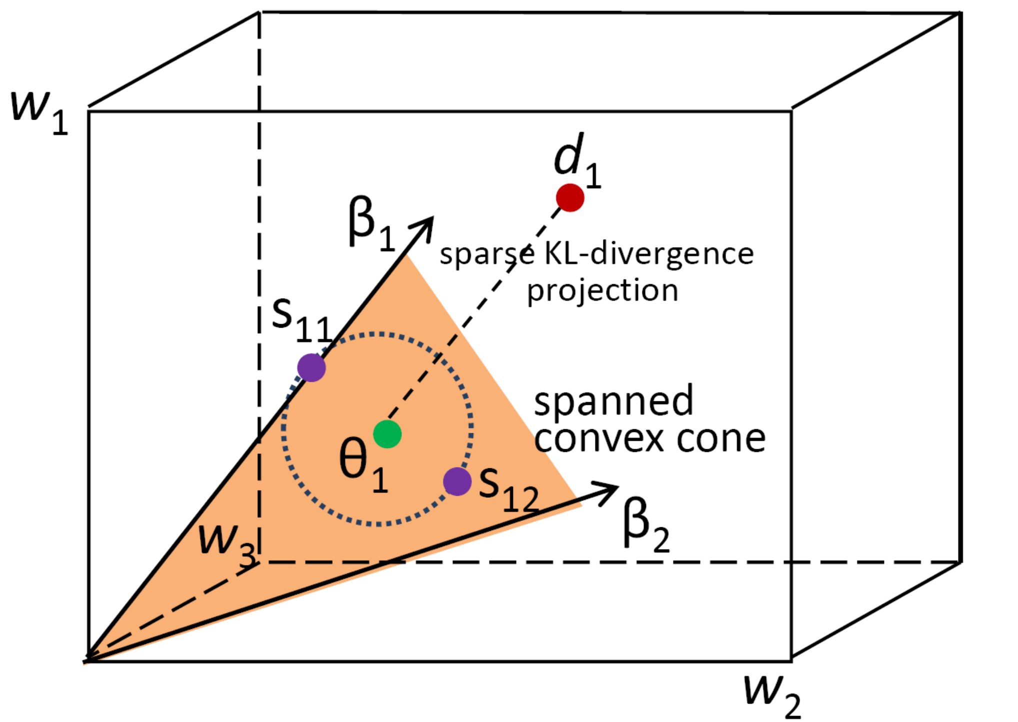
$$\min_{\{\mathbf{s}_{dn}, \theta_d\}, \beta} \sum_{d, n \in I_d} \ell(w_{dn}, \mathbf{s}_{dn}^\top \beta_{.n}) + \sum_{d, n \in I_d} \left(\frac{\gamma}{2}\|\mathbf{s}_{dn} - \theta_d\|_2^2 + \rho\|\mathbf{s}_{dn}\|_1\right) + \lambda \sum_d \|\theta_d\|_1$$

$$\text{s.t.}: \quad \beta_k \in \mathcal{P}_N, \ \forall k; \ \ \theta_d \geq 0, \ \forall d; \ \ \mathbf{s}_{dn} \geq 0, \ \forall d, n \in I_d;$$

Sparse Topical Coding (STC)

A Projection View

(unnormalized) KL-divergence for log-Poisson loss



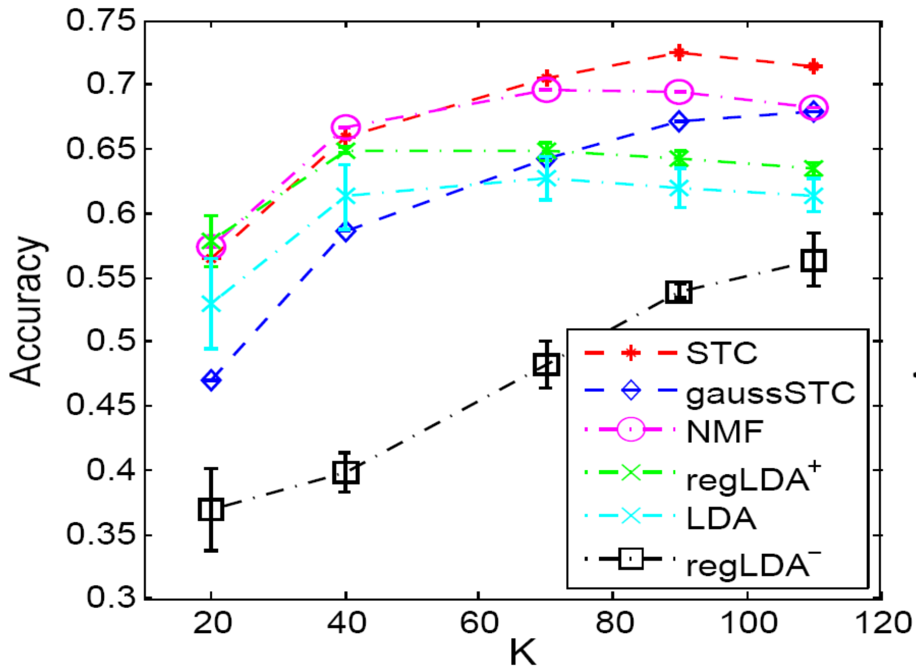
Projection is done under Regularization!

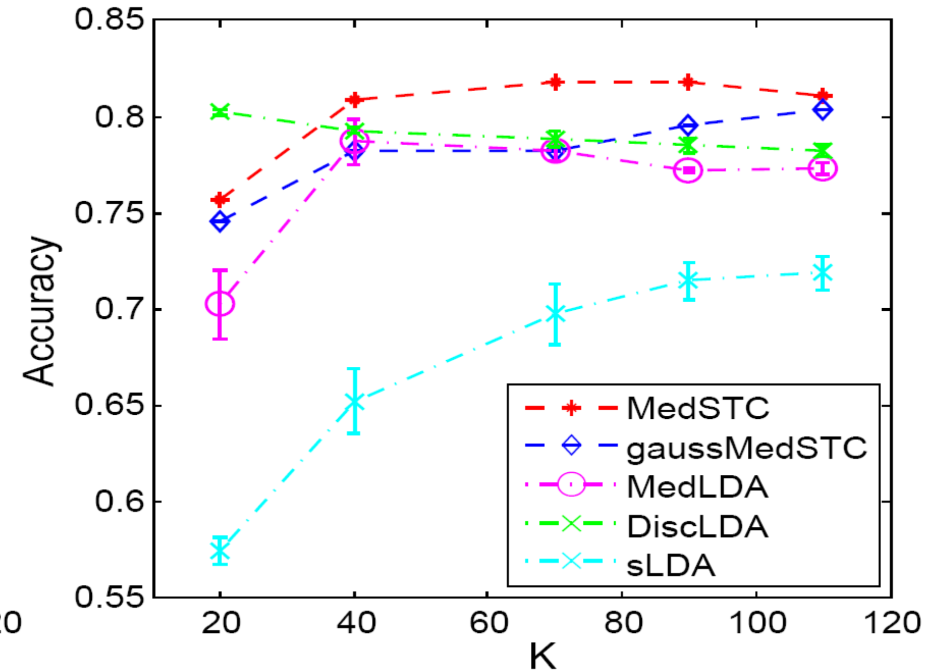
Experiments:
Sparse Topical Coding

- Data Sets:
  - 20 Newsgroups
  - Documents from 20 categories
  - ~ 20,000 documents in each group
  - Remove stop word as listed in UMASS Mallet

[Zhu & Xing, UAI 2011; Zhang & Zhu, WWW2013]

Prediction Accuracy on 20Newsgroups

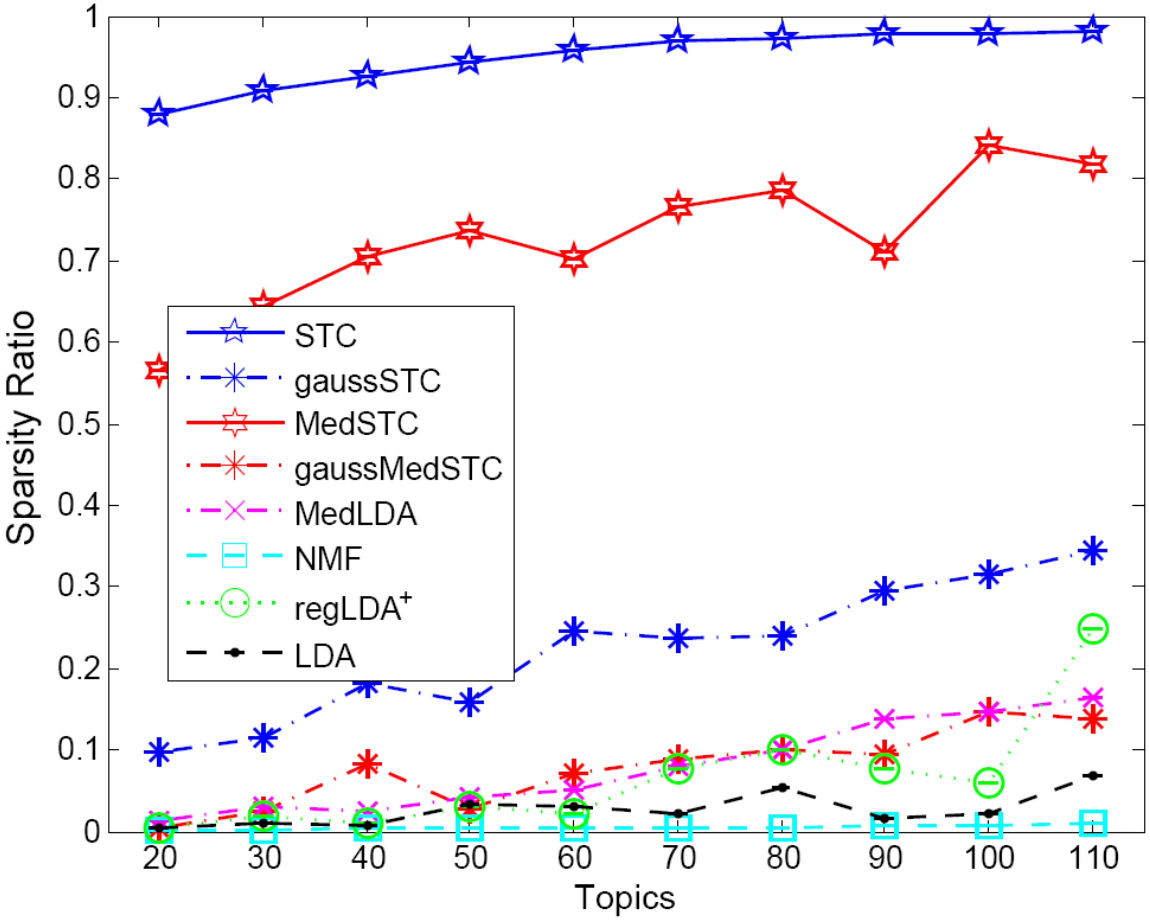




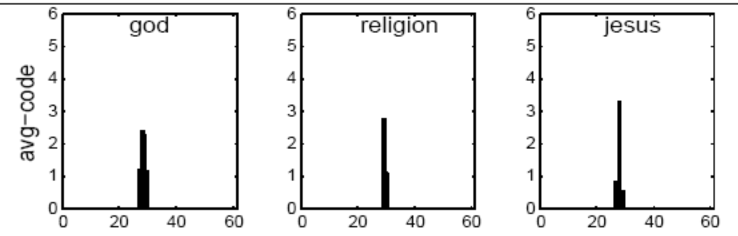
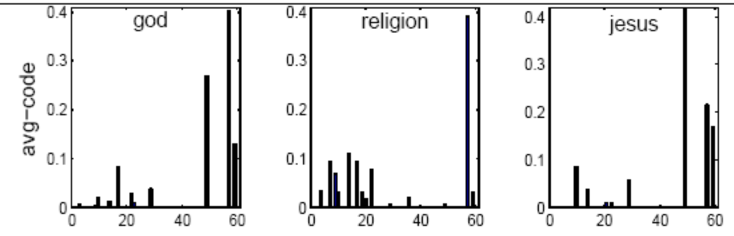
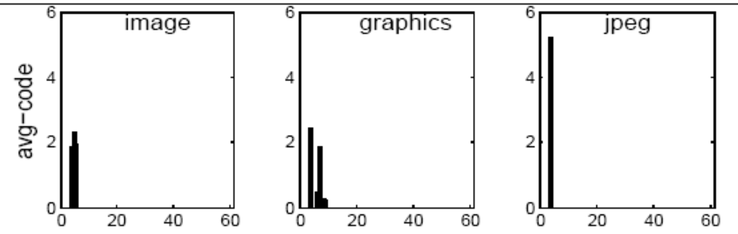
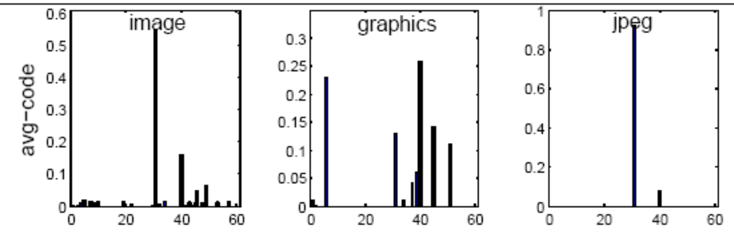
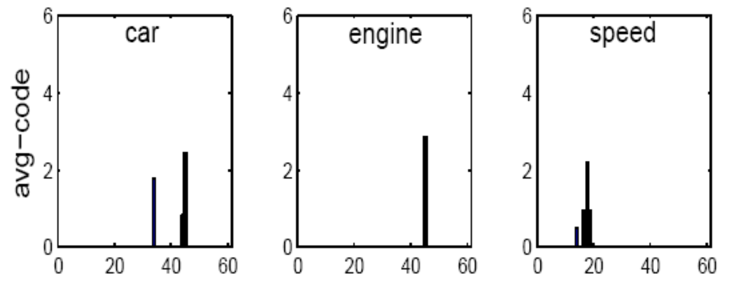
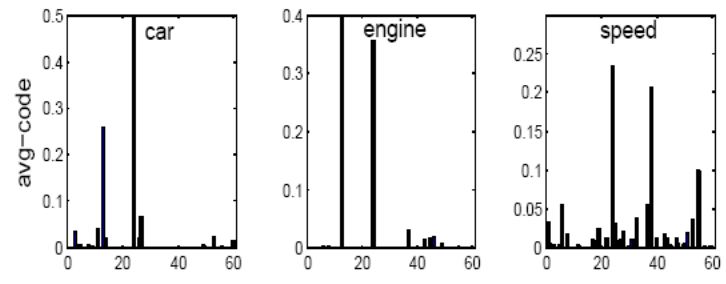
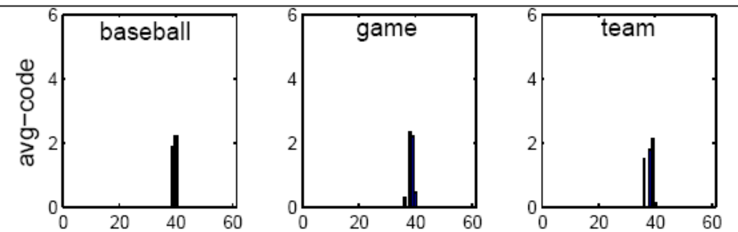
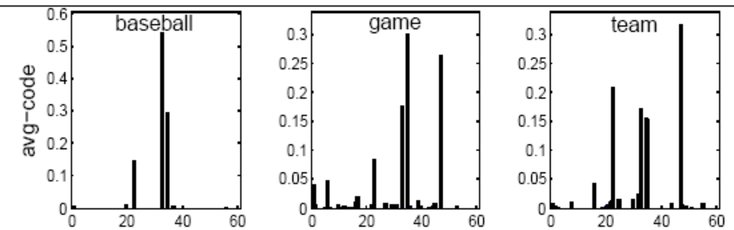
- gaussSTC: uses L2-norm regularizor on word and doc codes
- NMF: non-negative matrix factorization
- regLDA: LDA model using entropic regularizer on topic assignment distributions
- MedLDA: max-margin supervised LDA (Zhu et al., 2012)
- DiscLDA: discriminative LDA (Simon et al., 2008)

Sparsity of Word Codes on 20Newsgroups

- Sparsity ratio: the percentage of zero elements on the word codes



Sparse Word Codes on 20Newsgroups

Class	STC	LDA
alt. atheism		
comp. graphics		
rec. autos		
sport. baseball		

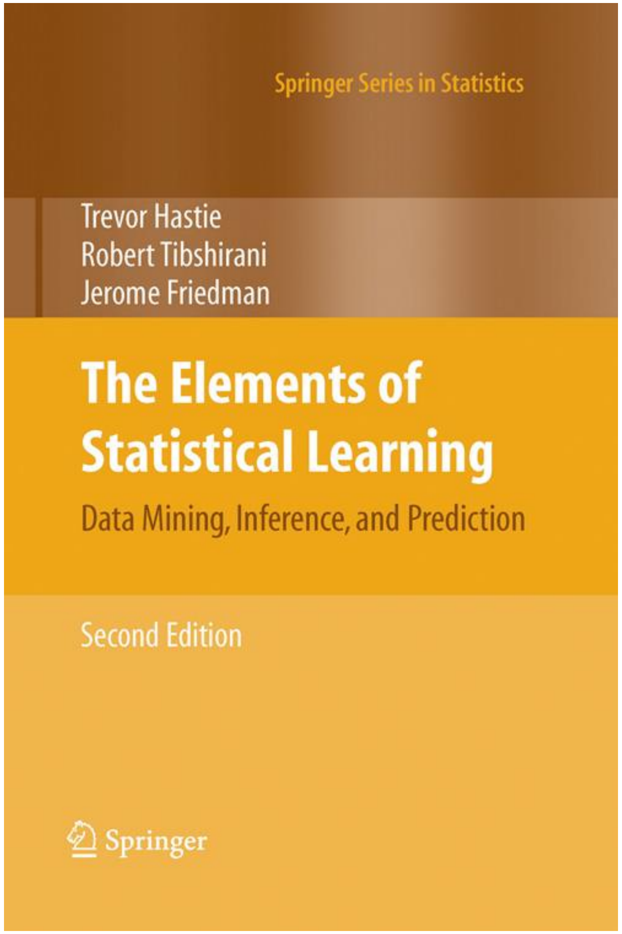
References

- Robert Tibshirani (1996), Regression Shrinkage and Selection via the Lasso, Journal of the Royal Statistical Society, Series B, 58, 267-288.
- Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani (2004), Least Angle Regression, Annals of Statistics, 32, 407-451.
- Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). A note on the group lasso and a sparse group lasso. Tech. Report.
- Francis Bach, Rodolphe Jenatton, Julien Mairal and Guillaume Obozinski. Optimization with Sparsity-Inducing Penalties. Foundations and Trends in Machine Learning, Vol. 4, No. 1 (2011)

- Software:
  - SPAMS (SPArse Modeling Software): http://www.di.ens.fr/willow/SPAMS/

Additional reading materials

- Chap. 3 of Elements of Statistical Learning (2nd Edition)
  - http://statweb.stanford.edu/~tibs/ElemStatLearn/



Additional reading materials

Foundations and Trends® in
Machine Learning
Vol. 4, No. 1 (2011) 1–106

DOI: 10.1561/2200000015

now
the essence of knowledge

Optimization with Sparsity-Inducing Penalties

By Francis Bach, Rodolphe Jenatton,
Julien Mairal and Guillaume Obozinski

Additional reading materials

Foundations and Trends® in Optimization
Vol. 1, No. 3 (2013) 123–231

DOI: xxx

the essence of knowledge

Proximal Algorithms

Neal Parikh
Department of Computer Science
Stanford University
npparikh@cs.stanford.edu

Stephen Boyd
Department of Electrical Engineering
Stanford University
boyd@stanford.edu