CS540 Machine learning
Lecture 13
L1 regularization

Outline

- L1 regularization
- Algorithms
- Applications
- Group lasso
- Elastic net

L1 regularized optimization

- Objective

$$
\begin{aligned}
J(\mathbf{w}) &= R(\mathbf{w}) + \lambda ||\mathbf{w}||_1 \\
R(\mathbf{w}) &= \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, \mathbf{w})) \\
L(y, \hat{y}) &= (y - \hat{y})^2 \\
L(y, \hat{y}) &= I(y \neq \hat{y})
\end{aligned}
$$

Lasso

The Lasso Page

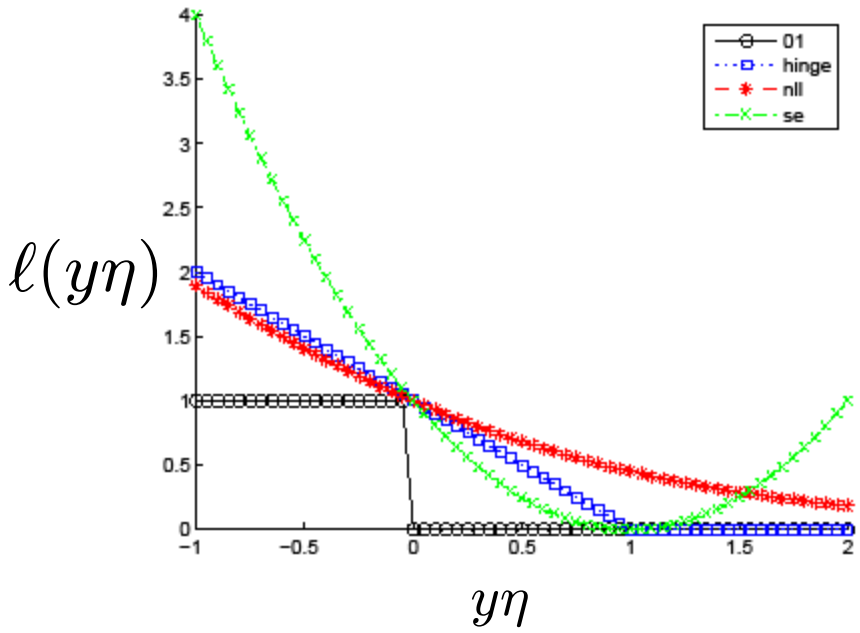
L1-constrained fitting
for statistics and data mining

$$J(\mathbf{w}) \quad = \quad RSS(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

Convex bounds to 0-1 loss

For y $\in$ {-1,+1}

$$
\begin{aligned}
p(y_i|\mathbf{x}_i, \mathbf{w}) &= \sigma(y_i \eta_i) \\
\eta_i &= \mathbf{w}^T \mathbf{x}_i = f(\mathbf{x}_i, \mathbf{w}) \\
L^{nll}(y, \eta) &= -\log p(y|\mathbf{x}, \mathbf{w}) = \log(1 + e^{-y\eta}) \\
L^{01}(y, \eta) &= I(y\eta < 0)
\end{aligned}
$$

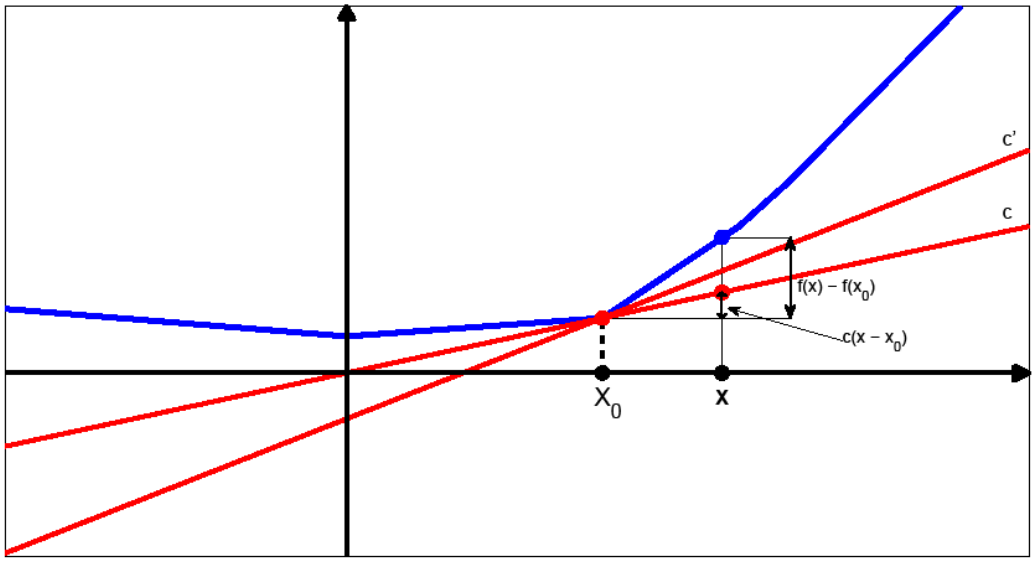


First order optimality conditions

- Objective is non differentiable at w=0, so cannot just require gradient = 0

$$J(\mathbf{w}) \quad = \quad R(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

Sub gradient



Subgradient $$f(x) \geq f(\mathbf{x}_0) + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_0) \forall \mathbf{x}$$

Subdifferential – convex set of sub gradients

$$\partial f(x) = \begin{cases} \{-1\} & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ \{+1\} & \text{if } x > 0 \end{cases}$$

f(x) = |x|

Optimality conditions

$$J(\mathbf{w}) \quad = \quad R(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

$$\partial_{w_j} J(\mathbf{w}) = \begin{cases} \{\nabla_j R - \lambda\} & \text{if } w_j < 0 \\ [\nabla_j R - \lambda, \nabla_j R + \lambda] & \text{if } w_j = 0 \\ \{\nabla_j R + \lambda\} & \text{if } w_j > 0 \end{cases}$$

At optimum, we must have

$$\begin{array}{cc} \nabla_j R(\mathbf{w}) + \lambda \text{sign}(w_j) = 0 & \text{if } |w_j| > 0 \\ |\nabla_j R(\mathbf{w})| \leq \lambda & \text{if } w_j = 0 \end{array}$$

Optimality conditions for L2 loss

$$\begin{aligned} \nabla_j R(\mathbf{w}) + \lambda \text{sign}(w_j) = 0 \quad & \text{if } |w_j| > 0 \\ |\nabla_j R(\mathbf{w})| \leq \lambda \quad & \text{if } w_j = 0 \end{aligned}$$

$$\begin{aligned} R(\mathbf{w}) &= \sum_i L(y_i, \mathbf{w}^T \mathbf{x}_i) \\ \nabla_j R(\mathbf{w}) &= \sum_i L'(y_i, \mathbf{w}^T \mathbf{x}_i) x_{ij} \\ L(y_i, \hat{y}_i) &= (y_i - \hat{y}_i)^2 \\ \nabla_j R(\mathbf{w}) &= 2\mathbf{X}(:, j)^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \end{aligned}$$

Optimality for L2 loss

$$J(\mathbf{w}) = RSS(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

- Homework

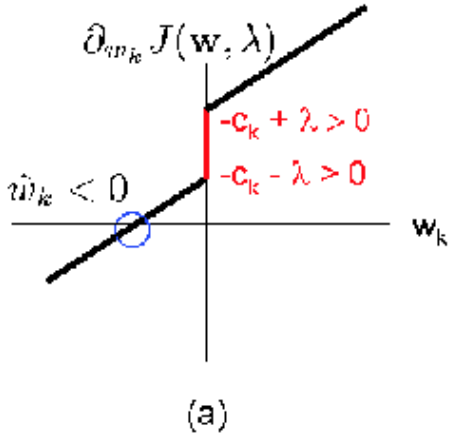
$$\begin{aligned}
\frac{\partial}{\partial w_k} RSS(\mathbf{w}) &= a_k w_k - c_k \\
a_k &= 2 \sum_{i=1}^{n} x_{ik}^2 \\
c_k &= 2 \sum_{i=1}^{n} x_{ik} (y_i - \mathbf{w}_{-k}^T \mathbf{x}_{i,-k}) \\
&= 2 \sum_{i=1}^{n} \left[x_{ik} y_i - x_{ik} \mathbf{w}^T \mathbf{x}_i + w_k x_{ik}^2 \right]
\end{aligned}$$

C_k = correlation between xk and r_{-k}

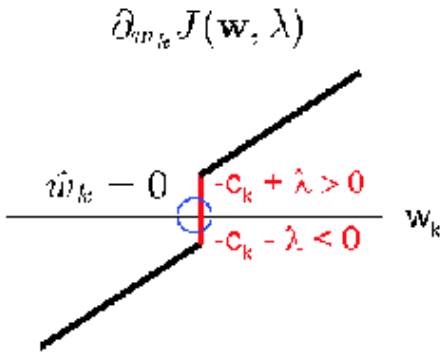
Subgradient

$$\partial_{w_k} J(\mathbf{w}, \lambda) = (a_k w_k - c_k) + \lambda \partial_{w_k} ||\mathbf{w}||_1$$

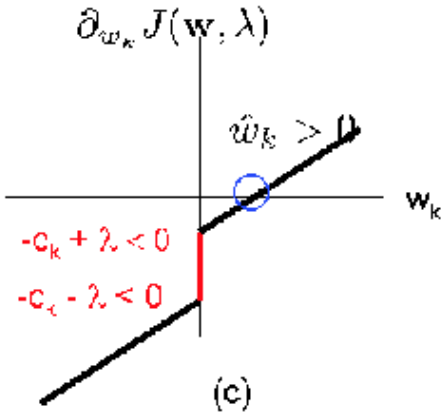
$$= \begin{cases} \{a_k w_k - c_k - \lambda\} & \text{if } w_k < 0 \\ [-c_k - \lambda, -c_k + \lambda] & \text{if } w_k = 0 \\ \{a_k w_k - c_k + \lambda\} & \text{if } w_k > 0 \end{cases}$$



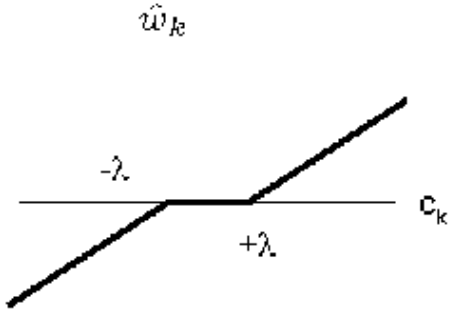
(a)



(b)



(c)



(d)

Soft thresholding

- We set weights for weakly correlated features to 0

$$\hat{w}_k(c_k) = \begin{cases} (c_k + \lambda)/a_k & \text{if } c_k < -\lambda \\ 0 & \text{if } c_k \in [-\lambda, \lambda] \\ (c_k - \lambda)/a_k & \text{if } c_k > \lambda \end{cases}$$

$$\hat{w}_k = sign(\frac{c_k}{a_k}) \max\{0, |\frac{c_k}{a_k}| - \frac{\lambda}{a_k}\} = \text{sign}(\frac{c_k}{a_k}) \left(|\frac{c_k}{a_k}| - \frac{\lambda}{a_k}\right)_+$$

- For orthonormal features

$$\hat{w}_k^{lasso} = \text{sign}(\hat{w}_k^{OLS}) \left(|\hat{w}_k^{OLS}| - \frac{\lambda}{2}\right)_+$$

$$\hat{w}_k^{ridge} = \frac{\hat{w}_k^{OLS}}{1 + \lambda}$$

$$\hat{w}_k^{SS} = \begin{cases} \hat{w}_k^{OLS} & \text{if rank}(|w_k|) \leq K \\ 0 & \text{otherwise} \end{cases}$$

Outline

- L1 regularization
- Algorithms
- Applications
- Group lasso
- Elastic net

Algorithms for lasso

- Subgradient methods
  - Gauss-Seidel, Grafting, Coordinate descent (shooting)

$$J(\mathbf{w}) = R(\mathbf{w}) + \lambda ||\mathbf{w}||_1 \qquad \mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \mathbf{g}_t, \mathbf{g}_t \in \partial J(\mathbf{w}_t)$$

- Constrained formulation

$$J(\mathbf{w}) = R(\mathbf{w}) \text{ s.t. } ||\mathbf{w}||_1 \leq t$$

  - QP, Interior point, Projected gradient descent
- Smooth unconstrained approximations
  - Approximate L1 penalty, use eg Newton's

$$||\mathbf{w}||_1 \approx \sum_j \sqrt{w_j + \epsilon}$$

Coordinate descent

- Coordinate descent (shooting: Fu, 1998)

$$\hat{w}_k(c_k) = \begin{cases} (c_k + \lambda)/a_k & \text{if } c_k < -\lambda \\ 0 & \text{if } c_k \in [-\lambda, \lambda] \\ (c_k - \lambda)/a_k & \text{if } c_k > \lambda \end{cases}$$

Listing 1: :

```
function [beta,iter] = LassoShooting(X, y, lambda,varargin)
[n p] = size(X);
iter = 0;
XX2 = X'*X*2;
Xy2 = X'*y*2;
converged = 0;
while ~converged & (iter < maxIter)
  beta_old = beta;
  for j = 1:p
    cj = Xy2(j) - sum(XX2(j,:)*beta) + XX2(j,j)*beta(j);
    aj = XX2(j,j);
    if cj < -lambda
      beta(j,1) = (cj + lambda)/aj;
    elseif cj > lambda
      beta(j,1) = (cj  - lambda)/aj;
    else
      beta(j,1) = 0;
    end
  end
  iter = iter + 1;
```

Quadratic programming

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 \text{ subject to } ||\mathbf{w}||_1 \leq t$$

$$w_1 - w_2 \leq t, \;\; w_1 + w_2 \leq t, \;\; -w_1 - w_2 \leq t, \;\; -w_1 + w_2 \leq t$$

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \leq \begin{pmatrix} t \\ t \\ t \\ t \end{pmatrix}$$

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} + \mathbf{f}^T\mathbf{w} \text{ subject to } \mathbf{A}\mathbf{w} \leq \mathbf{b}$$

$$\mathbf{H} = X^T\mathbf{X}, \;\; \mathbf{f} = -(\mathbf{y}^T\mathbf{X})^T, \;\; \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}, \;\; \mathbf{b} = t\mathbf{1}$$

Listing 1: :

```
wInit = X\y; % Start from the Least Squares solution
options = optimset('Display',display,'MaxIter',maxIter);
w = quadprog(X'*X,-y'*X,A,b,[],[],-t*ones(p,1),t*ones(p,1),wInit,options);
```

QP II

- Replace 2^d constraints with 2d

$$
\begin{aligned}
w_j &= w_j^+ - w_j^- \\
|w_j| &= w_j^+ + w_j^-, \;\; w_j^+ \geq 0, \;\; w_j^- \geq 0 \\
\sum_{j=1}^{d} |w_j| \leq t \;\; &\rightarrow \;\; \sum_{j=1}^{d} (w_j^+ + w_j^-) \leq t
\end{aligned}
$$

Listing 1: :

```
[n p] = size(X);
X = [X -X];
A = [-1*eye(2*p,2*p);ones(1,2*p)];
b = [zeros(2*p,1);t];
options = optimset('Display',display,'MaxIter',maxIter);
[w2] = quadprog(X'*X,-y'*X,A,b,[],[],-t*ones(2*p,1),t*ones(2*p,1),zeros(2*p,1),
    options);
w = [w2(1:p)-w2(p+1:2*p)];
```

Active set analysis for lasso

- 0 is a subgradient at w iff for all j
- Active variable condition

$$\text{sign}(w_j) \neq 0 \quad \Rightarrow \quad \mathbf{X}(:,j)^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \text{sign}(w_j) = 0$$

- Inactive variable condition

$$\text{sign}(w_j) = 0 \quad \Rightarrow \quad |\mathbf{X}(:,j)^T(\mathbf{y} - \mathbf{X}\mathbf{w})| \leq \lambda$$

- Let s_j in {-1,0,1}, J = {j: $s_j \neq 0$}

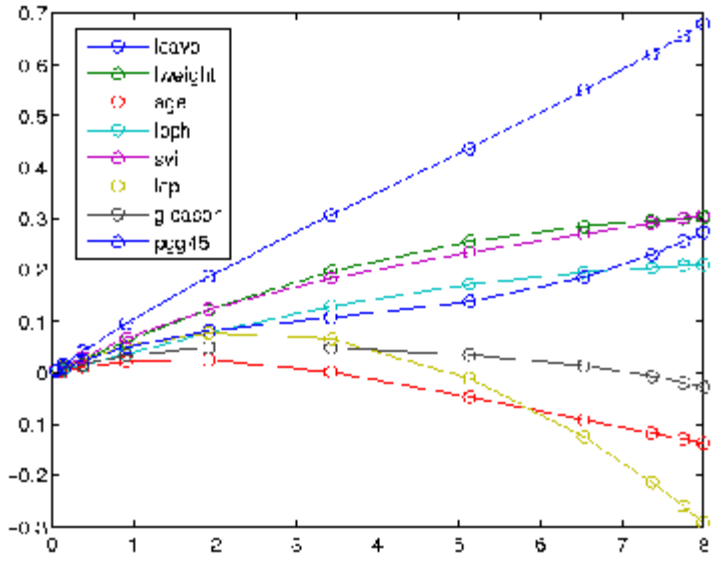
$$\mathbf{w}_J \quad = \quad (\mathbf{X}_J^T \mathbf{X}_J)^{-1}(\mathbf{X}_J^T \mathbf{y} + \lambda \mathbf{s}_J)$$

- Active set changes with lambda

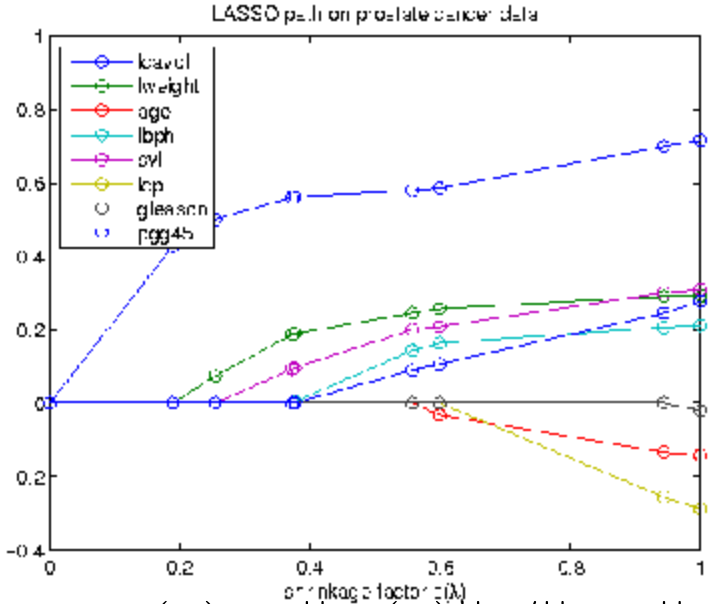
LARS

- Least Angle Regression and Shrinkage
- This is a way to find w for every point at which the active set changes as we sweep $\lambda=\lambda_{max}$ to $\lambda=0$
- Start at w=0
- Add variable most correlated with current residual
- Remove a variable from current active set if one of the optimality conditions is violated
- After adding min(n,d) variables, reaches OLS
- Takes O(nd^2 + d^3) time

Regularization path



dof(λ)



$s(\lambda) = ||\mathbf{w}(\lambda)||_1 / ||\mathbf{w}_{ls}||_1$

Listing 1: :

```
     0      0      0      0      0      0      0      0
0.4279      0      0      0      0      0      0      0
0.5015 0.0735      0      0      0      0      0      0
0.5610 0.1878      0      0 0.0930      0      0      0
0.5622 0.1890      0 0.0036 0.0963      0      0      0
0.5797 0.2456      0 0.1435 0.2003      0      0 0.0901
0.5864 0.2572 -0.0321 0.1639 0.2082      0      0 0.1066
0.6994 0.2910 -0.1337 0.2062 0.3003 -0.2565      0 0.2452
0.7164 0.2926 -0.1425 0.2120 0.3096 -0.2890 -0.0209 0.2773
```

Piecewise linearity

- Coefficient path is piecewise linear
- For each discrete point k, can find λ(k) that produces w(:,k)

$$\lambda = 2||\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})||_\infty$$

- Then can find w(j) for any λ by linearly interpolating between w(j,λ(k-1)) and w(j,λ(k))

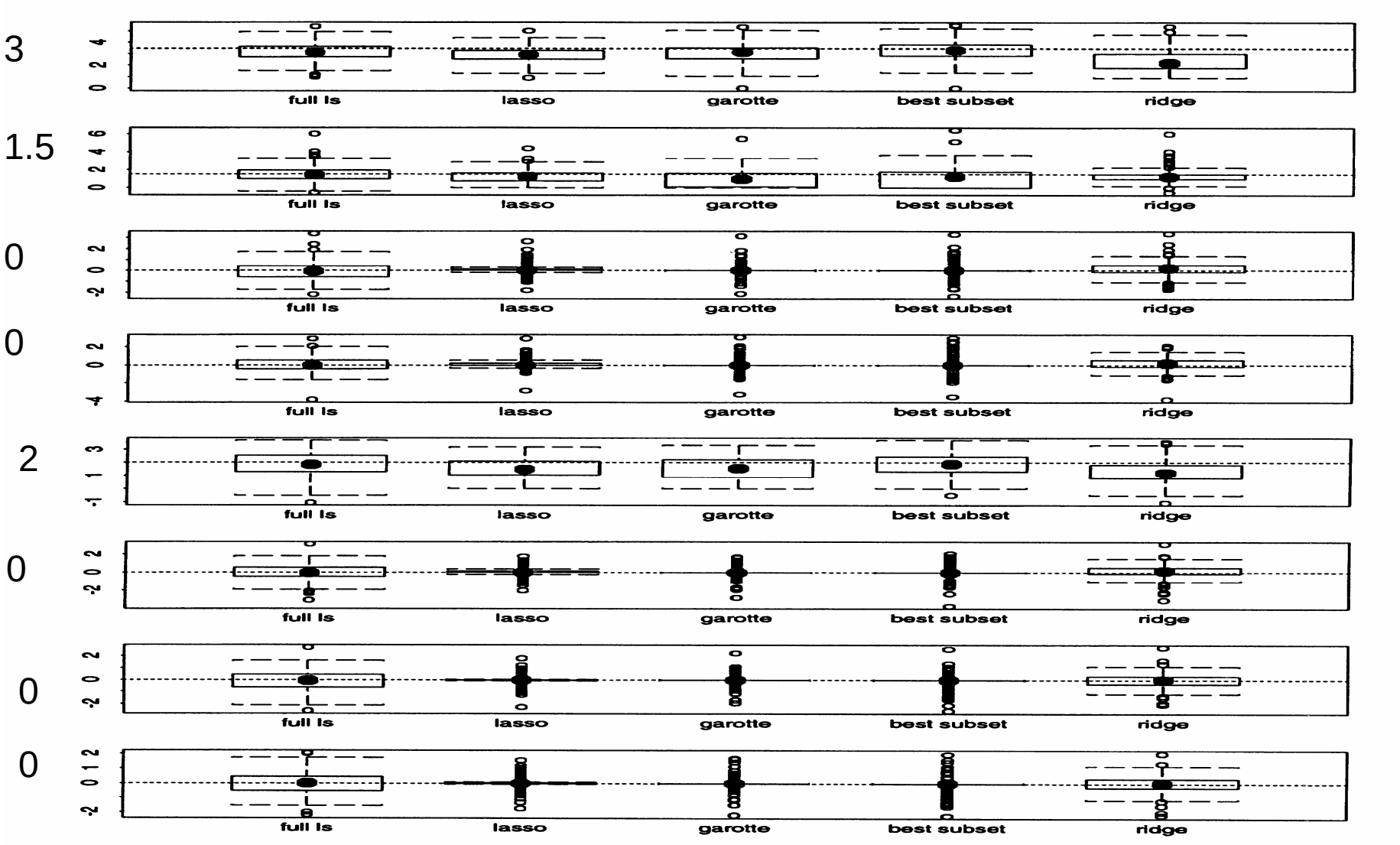
Outline

- L1 regularization
- Algorithms
- Applications
- Group lasso
- Elastic net

Variable selection

- We generate d random weights,
  of which z are zero nz are non-zero.
- We generate X (n x d) from correlated Gaussian
- We generate y = X w + noise
- We estimate w and compare the zero pattern to the truth

Coefficients



d=8 (z=5, nz=3), n=20, 50 trials

Tibshirani'96

MSE and sparsity

TABLE 3
Results for example 1†

Method	*Median mean-squared error*	*Average no. of 0 coefficients*	*Average ŝ*
Least squares	2.79 (0.12)	0.0	—
Lasso (cross-validation)	2.43 (0.14)	3.3	0.63 (0.01)
Lasso (Stein)	2.07 (0.10)	2.6	0.69 (0.02)
Lasso (generalized cross-validation)	1.93 (0.09)	2.4	0.73 (0.01)
Garotte	2.29 (0.16)	3.9	—
Best subset selection	2.44 (0.16)	4.8	—
Ridge regression	3.21 (0.12)	0.0	—

†Standard errors are given in parentheses.

d=8 (z=5, nz=3), n=20, 50 trials

Larger example

TABLE 8

Results for example 4†

Method	*Median mean-squared error*	*Average no. of 0 coefficients*	*Average ŝ*
Least squares	137.3 (7.3)	0.0	—
Lasso (Stein)	80.2 (4.9)	14.4	0.55 (0.02)
Lasso (generalized cross-validation)	64.9 (2.3)	13.6	0.60 (0.88)
Garotte	94.8 (3.2)	22.9	—
Ridge regression	57.4 (1.4)	0.0	—

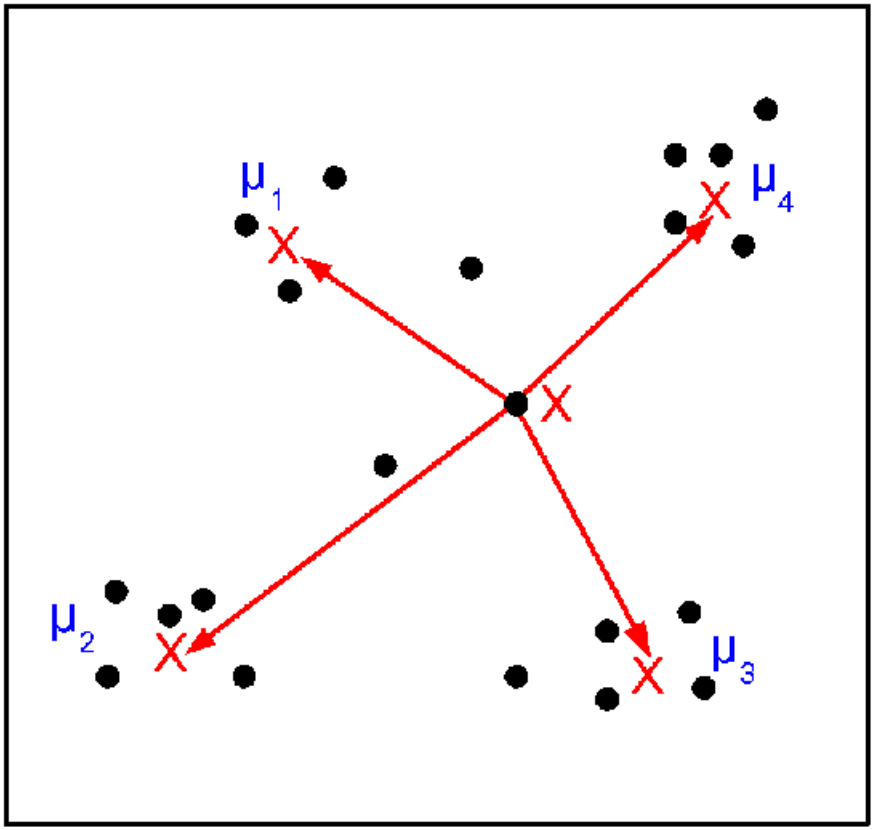
†Standard errors are given in parentheses.

d=40 (z=20, nz=20), n=100, 50 trials

"Kernelized" logistic regression

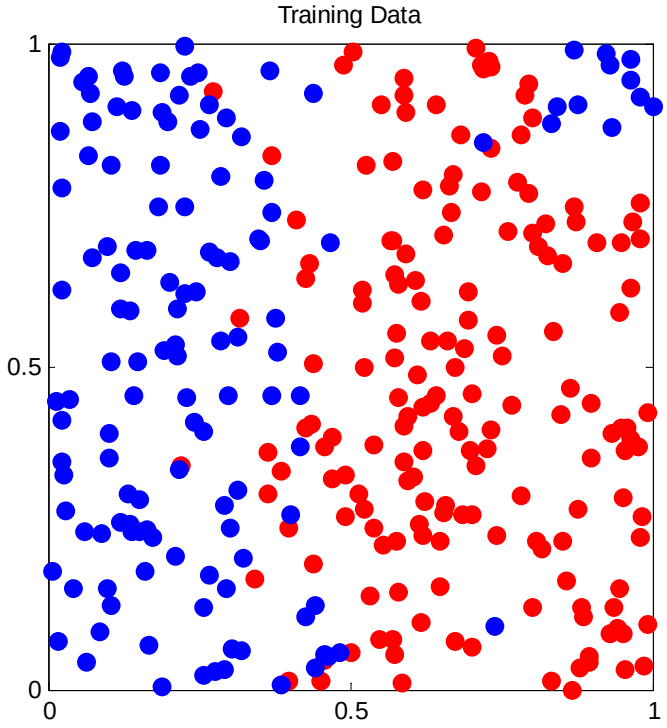
- Consider logistic regression with an RBF basis

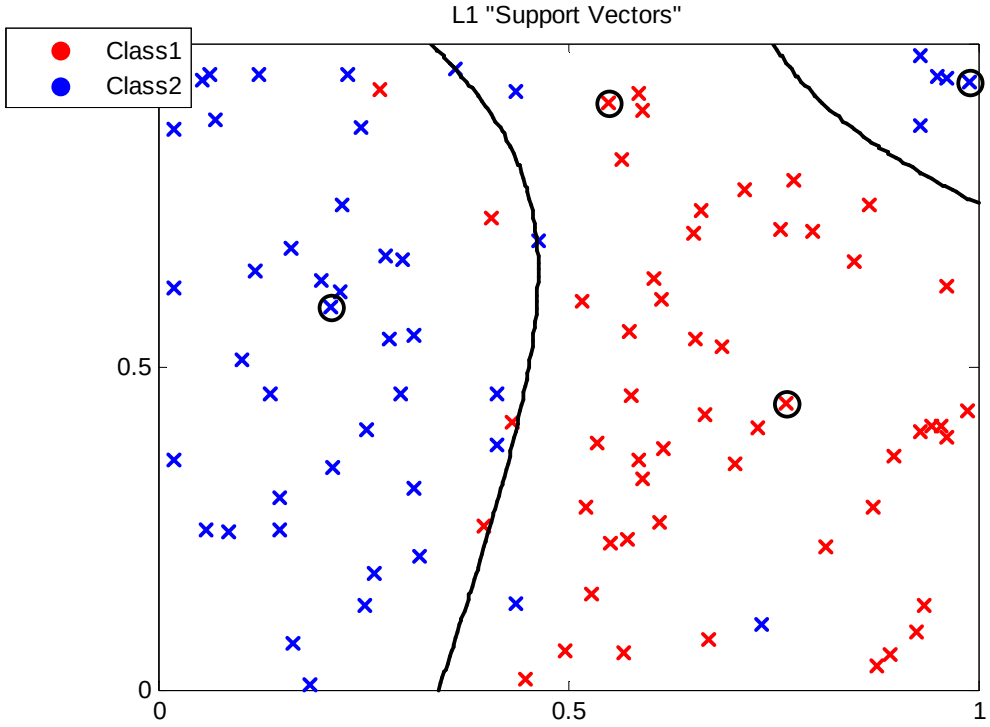
$$
\begin{aligned}
p(y|\mathbf{x},\mathbf{x}) &= \text{Ber}(y|\sigma(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}))) \\
\boldsymbol{\phi}(\mathbf{x}) &= [K(\mathbf{x},\boldsymbol{\mu}_1),\ldots,K(\mathbf{x},\boldsymbol{\mu}_d)] \\
K(\mathbf{x},\mathbf{x}') &= \mathcal{N}(\mathbf{x}|\mathbf{x}',\sigma^2\mathbf{I})
\end{aligned}
$$



Sparse kernelized logreg

- Now set $\mu_i = x_i$, and use an L1 penalty to select a subset of the basis functions





SMLR

	Crabs	Iris	FGlass	AML/ALL	Colon	Yeast	Mines
Number of classes (m)	2	3	6	2	2	5	2
Number of features	5	4	9	7129	2000	79	106
Total number of samples ($n + N$)	200	150	214	72	62	208	22933
Number of training samples (n)	80	135	193	38	50	182	11467
Number of test samples (N)	120	15	21	34	12	26	11466
k-fold cross-validation	—	10	10	—	30†	8	—

† In this case, we mean not 30-fold cross-validation but 30 different random 50/12 training/test splits.

	Crabs RBF		Iris RBF		FGlass RBF		AML/ALL linear		Colon linear		Yeast linear	
SVM	2	(53)	5	(124)	62	(365)	3	(31)	75	(32)	7	(155)
RVM	0	(4)	10	(37)	61	(74)	5	(3)	84	(6)	12	(49)
SMLR (l_1)	0	(10)	1	(136)	50	(901)	2	(10)	75	(15)	3	(83)
RMLR (l_2)	1	(80)	8	(270)	59	(965)	2	(38)	84	(50)	9	(728)
out of	120	(80)	150	(270)	214	(965)	34	(38)	360	(50)	208	(728)

Krishnapuram et al, 2005

Outline

- L1 regularization
- Algorithms
- Applications
- Group lasso
- Elastic net

Group lasso

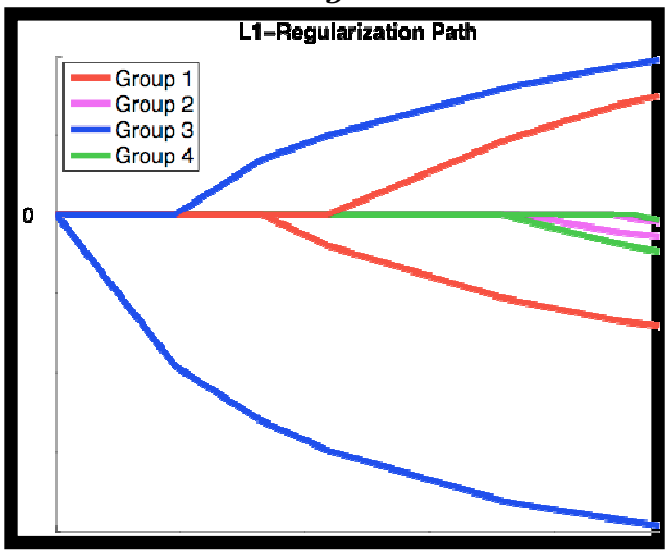
- Sometimes we want to select groups of parameters together (e.g., when encoding categorical inputs)

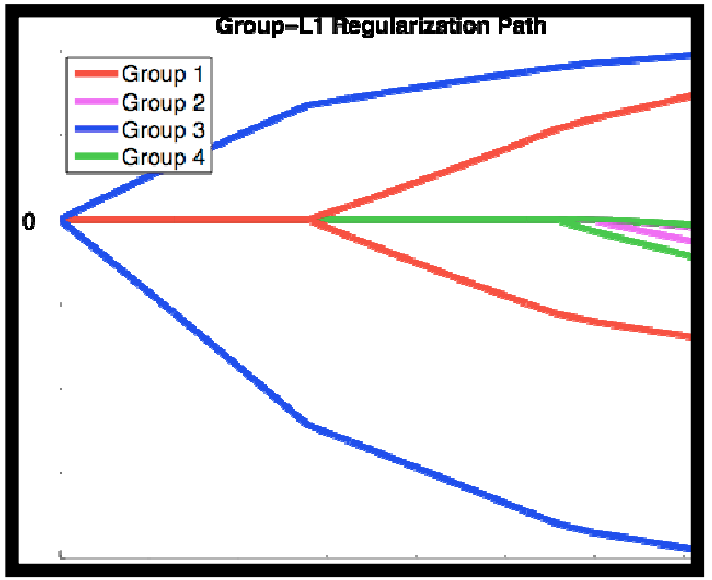
$$\hat{\mathbf{w}} = \arg\min R(\mathbf{w}) + \lambda f(\mathbf{w})$$

$$f(\mathbf{w}) = \sum_g ||\mathbf{w}_g||_2 = \sum_g \sqrt{\sum_{j \in g} w_{gj}^2}$$

$$f(\mathbf{w}) = \sum_g ||\mathbf{w}_g||_\infty = \sum_g \max_{j \in g} |w_{gj}|$$

Still convex, but much harder to optimize…





Elastic net

- Stretchable fishing net that retains all the big fish

$$J(\mathbf{w}, \lambda_1, \lambda_2) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda_2||\mathbf{w}||_2^2 + \lambda_1||\mathbf{w}||_1$$

- L1 on modified data

$$\begin{aligned}
\tilde{\mathbf{X}} &= c\begin{pmatrix}\mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I}_d\end{pmatrix}, \quad \tilde{\mathbf{y}} = \begin{pmatrix}\mathbf{y} \\ \mathbf{0}_{d\times 1}\end{pmatrix} \\
\tilde{\mathbf{w}} &= \arg\min_{\tilde{\mathbf{w}}} ||\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}||^2 + c\lambda_1||\tilde{\mathbf{w}}||_1 \\
c &= (1+\lambda_2)^{-\frac{1}{2}}
\end{aligned}$$

Xtilde has rank d, even if n < d, so can find d predictors.
Penalty is strictly convex so has the ‘grouping effect’: will select
All perfectly correlated features