RIDGE and LASSO regularization for regression

Feature selection

- Some algorithms perform naturally feature selection
  - for example Decision Trees, Boosting
- Other algorithms have difficulty with correlated features
  - for example Naive Bayes, Regression
- Some algorithms have difficulty with too many features

Feature selection

- Task(label) Independent, Model independent
  - Dimensionality reduction, clustering
  - PCA
- Filter Methods: Task dependent, Model independent
  - compute correlation among pairs of features
  - compute correlation of feature with labels
- Wrapper methods: Task dependent, Model dependent
  - try subsets of features with a given ML algorithm, pick a “best” subset

Forward Feature Selection

- Task dependent, Model dependent
- Select one feature at a time, dynamically
  - depending on how previous features do

set of features initial empty, $S = \emptyset$
repeat while improvement $> \epsilon$
    for each feature $f \notin S$
        performance $(S \cup \{f\})$ = performance(model, trained on $S \cup \{f\}$)
    end for
    $f_{new} = \text{argmax}_f$ performance$(S \cup \{f\})$
    improvement = performance $(S \cup \{f_{new}\})$ - performance (S)
    $S = S \cup \{f_{new}\}$
end repeat

Problems with regression

- Free coefficients (unconstrained) can result in problems
  - features canceling each other
  - features overwhelming each other
  - large complexity with no generalization benefit
- Solution : constrain the coefficients

Regularization for regression

- Regression: same as before, a linear predictor

$$h_w(x) = w^0 + x^T w = w^0 + \sum_j x^j w^j$$

- Regularized regression means add a "complexity" penalty in the objective
  - the objective contains the traditional least square (to be minimized)
  - but also R(w) a notion of complexity (to be minimized)

- λ tradeoffs the complexity for the objective

$$\min_{(w_0, w) \in \mathcal{R}^{d+1}} [\frac{1}{2N} \sum_{i=1}^{N} (y_i - w_0 - x_i^T w)^2 + \frac{\lambda}{N} R(w)]$$

Regularization for regression

- RIDGE penalty : L2 norm
  - causes all w coefficients to be small

$$L_2 \text{ norm } R(w) = \frac{1}{2}||w||_2^2 = \frac{1}{2}\sum_{j=1}^{d} w^{j2}$$

- LASSO penalty: L1 norm
  - causes some coefficients to be 0 (feature selection)

$$L_1 \text{ norm } R(w) = ||w||_1 = \sum_{j=1}^{d} |w^j|$$

- "elastic-net" : mixture of L1 and L2 norms

$$R_\beta(w) = (1-\beta)\frac{1}{2}||w||_2^2 + \beta||w||_1 = \sum_{j=1}^{d}[\frac{1}{2}(1-\beta)w^{j2} + \beta|w^j|]$$

Digits dataset

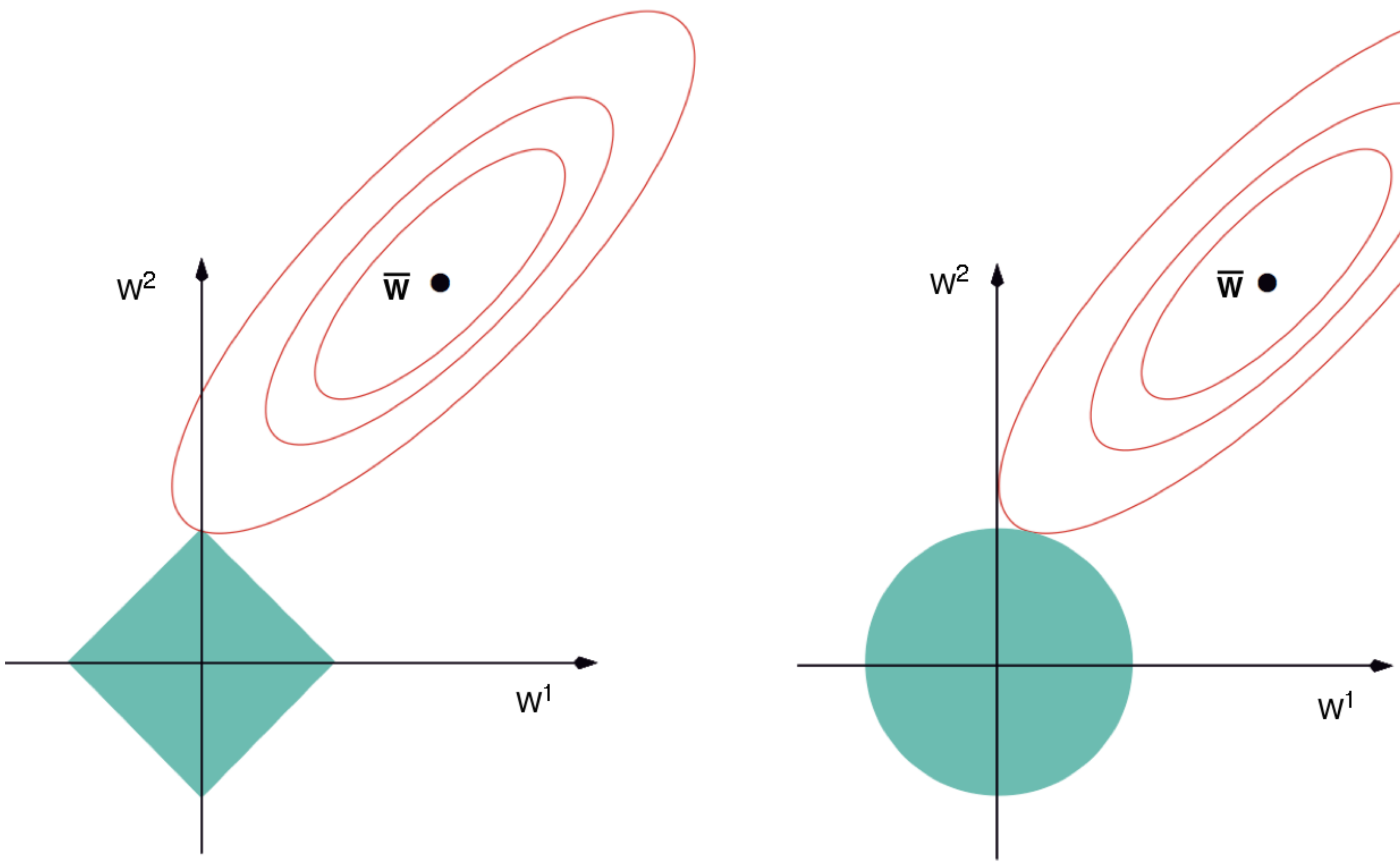
- can be written as constrained optimization
  - a direct correspondence between λ and t
  - solved by taking derivatives with Lagrangian Multipliers

$$\hat{w}^{ridge} = \arg\min_{w} \sum_{i=1}^{N} (y_i - w^0 - \sum_{j=1}^{d} x_i^j w^j)^2$$

$$\text{subject to } \sum_{j=1}^{d} w^{j2} \leq t$$

RIDGE vs LASSO

Figure 1: Source: Figure 3.11 of [4] Estimation picture for LASSO (left) and RIDGE (right). Solid areas are for regions of constraints $|w^1|+|w^2| \leq t$ (LASSO, left) and $(w^1)^2+(w^2)^2 \leq t$ (RIDGE, right). Red ellipses are the contours of the objective, here the least square function.



- the solution w will be in the feasible region (solid blue)

RIDGE vs LASSO

- RIDGE penalty for linear regression is essentially a regression problem with bigger matrices
  - Z = matrix data; n=number of data points, p=number of dimensions/features

The ℓ_2 criterion is the RSS for the augmented data set:

$$\mathbf{Z}_\lambda = \begin{pmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{n,1} & z_{n,2} & z_{n,3} & \cdots & z_{n,p} \\ \sqrt{\lambda} & 0 & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda} & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{\lambda} & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \sqrt{\lambda} \end{pmatrix} ; \; \mathbf{y}_\lambda = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

So:

$$\mathbf{Z}_\lambda = \begin{pmatrix} \mathbf{Z} \\ \sqrt{\lambda}\mathbf{I}_p \end{pmatrix} \; \mathbf{y}_\lambda = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}$$

  - like regression, admits analytical solution

$$\begin{aligned} (\mathbf{Z}_\lambda^\top \mathbf{Z}_\lambda)^{-1}\mathbf{Z}_\lambda^\top y_\lambda &= \left((\mathbf{Z}^\top, \sqrt{\lambda}\mathbf{I}_p) \begin{pmatrix} \mathbf{Z} \\ \sqrt{\lambda}\mathbf{I}_p \end{pmatrix} \right)^{-1} (\mathbf{Z}^\top, \sqrt{\lambda}\mathbf{I}_p) \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} \\ &= (\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top\mathbf{y}, \end{aligned}$$

RIDGE vs LASSO

- LASSO does not have an analytical solution
- RIDGE regularized regression can be solved with Gradient Descent : simply add a term to the gradient
  - same for RIDGE-Logistic regression
- LASSO can be solved via quadratic programming
  - or via approximation schemas like “forward stagewise”

Logistic Regression with RIDGE

$$h_w(\mathbf{x}) = g(w\mathbf{x}) = \frac{1}{1+e^{-w\mathbf{x}}} = \frac{1}{1+e^{-\sum_d w^d x^d}}$$

- like before, Logistic Regression optimizes max log likelihood of data
  - but now we add the L2 RIDGE penalty

$$\max_{w_0,w} \frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(P(y=1|x_i) + (1-y_i)\log(P(y=0|x_i)\right] - \frac{\lambda}{N}R(w)$$

$$\max_{w_0,w} \frac{1}{N}\sum_{i=1}^{N}\left[y_i \log h_w(x) + (1-y_i)\log(1-h_w(x))\right] - \frac{\lambda}{2N}\sum_{j=1}^{d} w^{j2}$$

- to use Gradient Descent we differentiate for each component j
  - gradient same as the one for logistic regression, except adding the differential of RIDGE penalty

$$\frac{\delta J}{\delta w^j} = \frac{1}{N}\sum_{i=1}^{N}(y_i - h_w(x_i))x_i^j + \frac{\lambda}{N}w^j$$

Logistic Regression with RIDGE

- The differential gives the Gradient Descend rule

$$w^0 := w^0 - \alpha \frac{1}{N} \sum_{i=1}^{N} (h_w(x_i) - y_i) x_i^0$$

$$w^j := w^j - \alpha \Big[\frac{1}{N} \sum_{i=1}^{N} (h_w(x_i) - y_i) x_i^j + \frac{\lambda}{N} w^j \Big] \quad \texttt{for any j = 1:d}$$