CMSC 510 – L13
Regularization Methods for Machine Learning

Instructor:
Dr. Tom Arodz

Regularization

- In general, we can introduce a penalty Ω over the space of classifiers
- Regularized empirical risk minimization:
  empirical risk of h() + penalty based on form of h()
  (based on training data) (based on assumptions, not on training data)

$$h^* = \arg\min_h \hat{R}_{S_m}(h) + \lambda\Omega(h)$$

  - Specifically, for linear classifiers $h(x)=w^T x$

$$w^* = \arg\min_w \hat{R}_{S_m}(w) + \lambda\Omega(w)$$

$$\widehat{R}_{S_m}(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h, \mathbf{z}_i)$$

$$\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$$

$$\mathcal{Z} = \mathcal{X} \times \{-1, +1\}$$

- What mathematical properties should we enforce on our choices of Ω?

Regularization

- Regularized empirical risk minimization:
  empirical risk of h() + penalty based on form of h()
  (based on training data) (based on assumptions, not on training data)

$$h^* = \arg\min_h \hat{R}_{S_m}(h) + \lambda\Omega(h)$$

  - Specifically, for linear classifiers $h(x)=w^Tx$

$$w^* = \arg\min_w \hat{R}_{S_m}(w) + \lambda\Omega(w)$$

$$\widehat{R}_{S_m}(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h, \mathbf{z}_i)$$

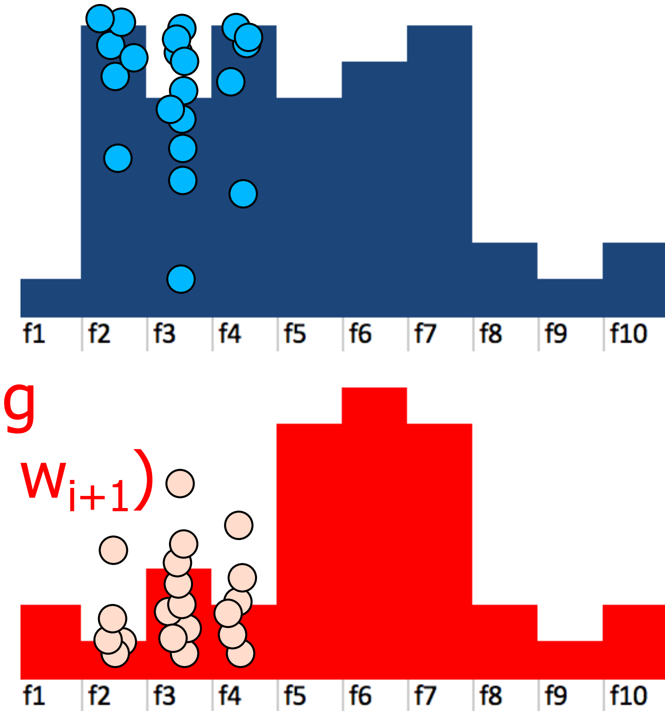
$$\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$$

$$\mathcal{Z} = \mathcal{X} \times \{-1, +1\}$$

- If the regularized classification problem is to have no local minima, we want Ω to be convex!

Fused Lasso

- Desired penalty structure:
  - If f_i is important for classification, f_{i-1} and f_{i+1} likely to be important, too
  - What we would like to have is a classifier where either both neighboring features are selected (non-zero w_i and w_{i+1}) or both are not selected ($w_i=w_{i+1}=0$)
    - That's impossible, unless all features are selected or no feature is selected
    - But we prefer fewer changes between neighbors

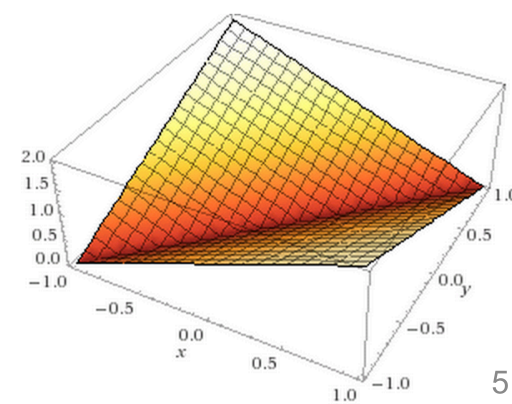


Fused L_1

- Regularized empirical risk minimization (e.g. logistic regression): $w^* = \arg\min_w \hat{R}_{S_m}(w) + \lambda\Omega(w)$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}|$$

- *Fused* L_1 penalty (*fused lasso, fused logistic regression*, etc.)
  - Sometimes also called: *total variation*
- Here's the shape of $\Omega(w)$ if we have two features
- $\Omega(w)$ is sum of convex $|w_f - w_{f-1}|$ => Ω is convex
  Why term $|w_f - w_{f-1}|$ is convex?
  - If
    h(w) and g(w) are convex,
    then
    f(w)=max(h(w),g(w)) is convex
  - What is f, h & g here?

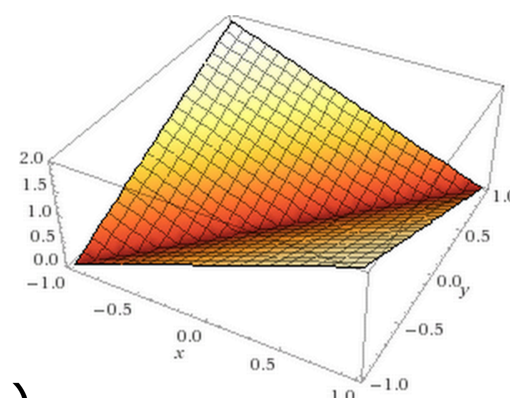


Fused L_1 regularization

- Regularized empirical risk minimization (e.g. logistic regression): $w^* = \arg\min_w \hat{R}_{S_m}(w) + \lambda\Omega(w)$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}|$$

- Fused L_1 penalty (fused lasso, fused logistic regression, etc.)
  - Sometimes also called: total variation
- $\Omega(w)$ is convex. Why?
  - If h(w) and g(w) are convex, then f(w)=max(h(w),g(w)) is convex
  - $h(w)=w_f - w_{f-1}$ linear => convex
  - $g(w)= -(w_f - w_{f-1}) = -h(w)$ linear => convex
  - $f(w) = |w_f - w_{f-1}| = |h(w)| = \max(h(w), -h(w))$
    - h(w) and –h(w) convex => f(w) convex => $\Omega(w)$ convex

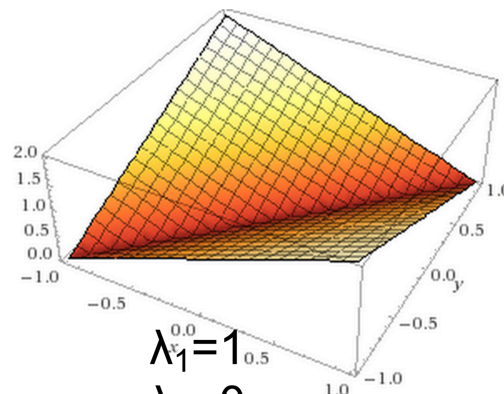


Fused L_1 regularization

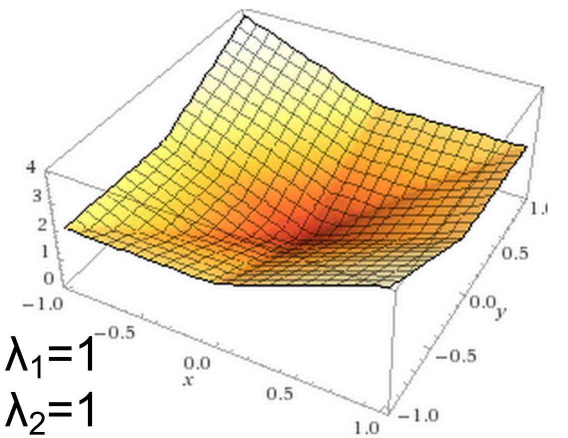
- Fused L_1 regularized empirical risk minimization (e.g. logistic regression):

$$w^* = \arg\min_w \hat{R}_{S_m}(w) + \lambda_1 \sum_{f=2}^{F} |w_f - w_{f-1}| + \lambda_2 \sum_{f=1}^{F} |w_f|$$

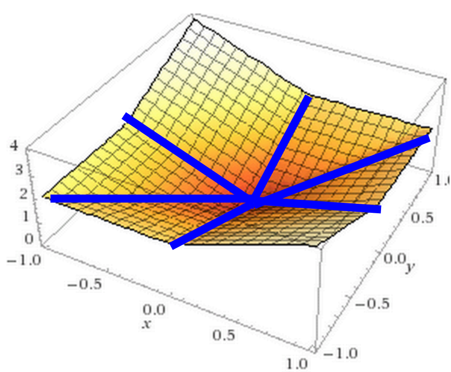
- How to obtain minimum of regularized risk?

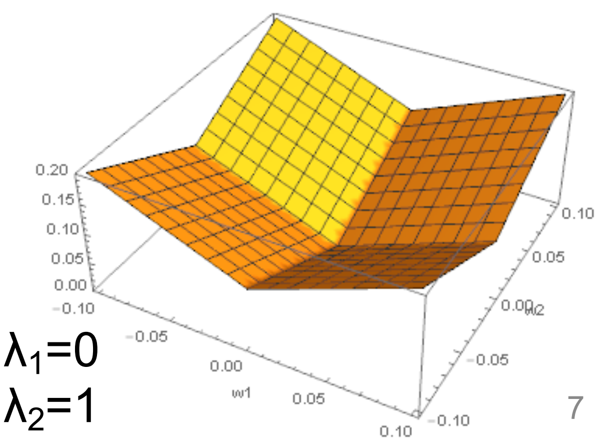


$\lambda_1=1$
$\lambda_2=0$



$\lambda_1=1$
$\lambda_2=1$





$\lambda_1=0$
$\lambda_2=1$

  - Non-differentiable

Ω = fused L_1 penalty + L_1 penalty

Problem: minimize regularized empirical risk: $R_S(w) + \Omega(w)$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

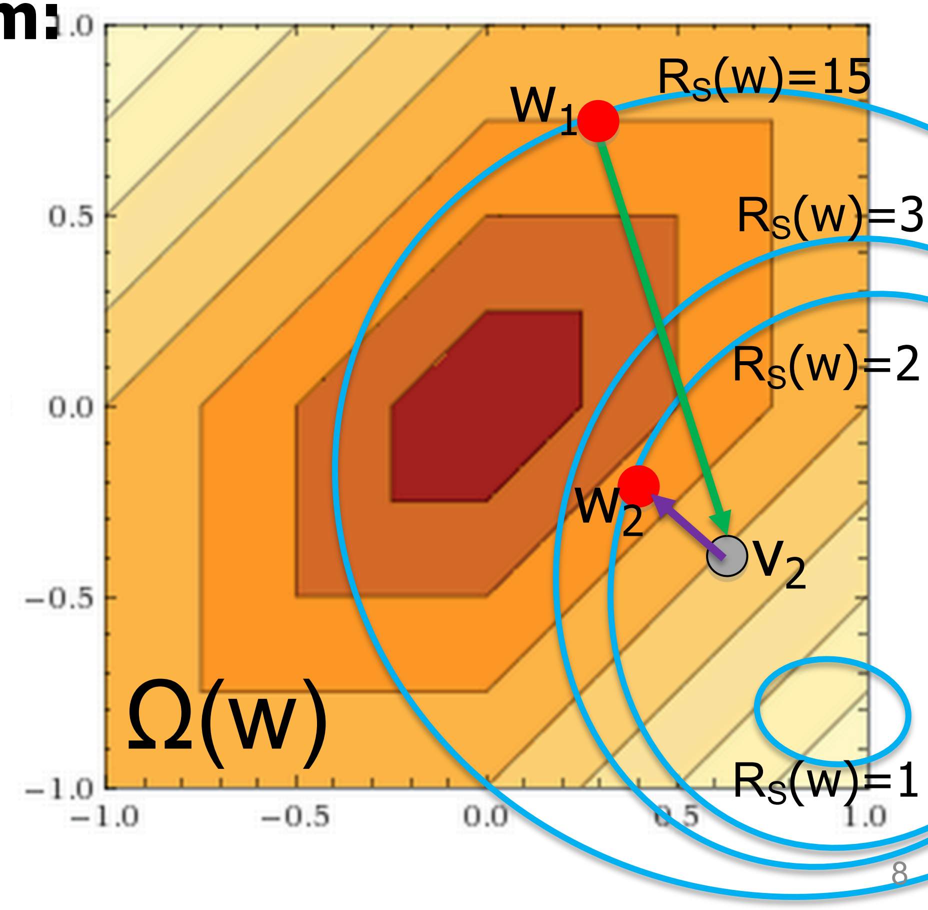
MM iterations toward minimum:

Gradient step:

$v_{n+1} = w_n - \nabla R_S(w_n)/L$

Proximal step:

$w_{n+1} = \mathrm{prox}_{\Omega,L/2}(v_{n+1})$



$\mathbf{prox}_{Q,b}(z) =$
$\mathrm{argmin}_x\, Q(x)+b||x - z||^2$

Proximal gradient method

Problem: minimize convex $R(\Phi) + \Omega(\Phi)$

Proximal operator: $\text{prox}_{\Omega,b}(\Psi) = \text{argmin}_{\Phi}\, \Omega(\Phi)+b||\Phi - \Psi||^2$

MM iteration:

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

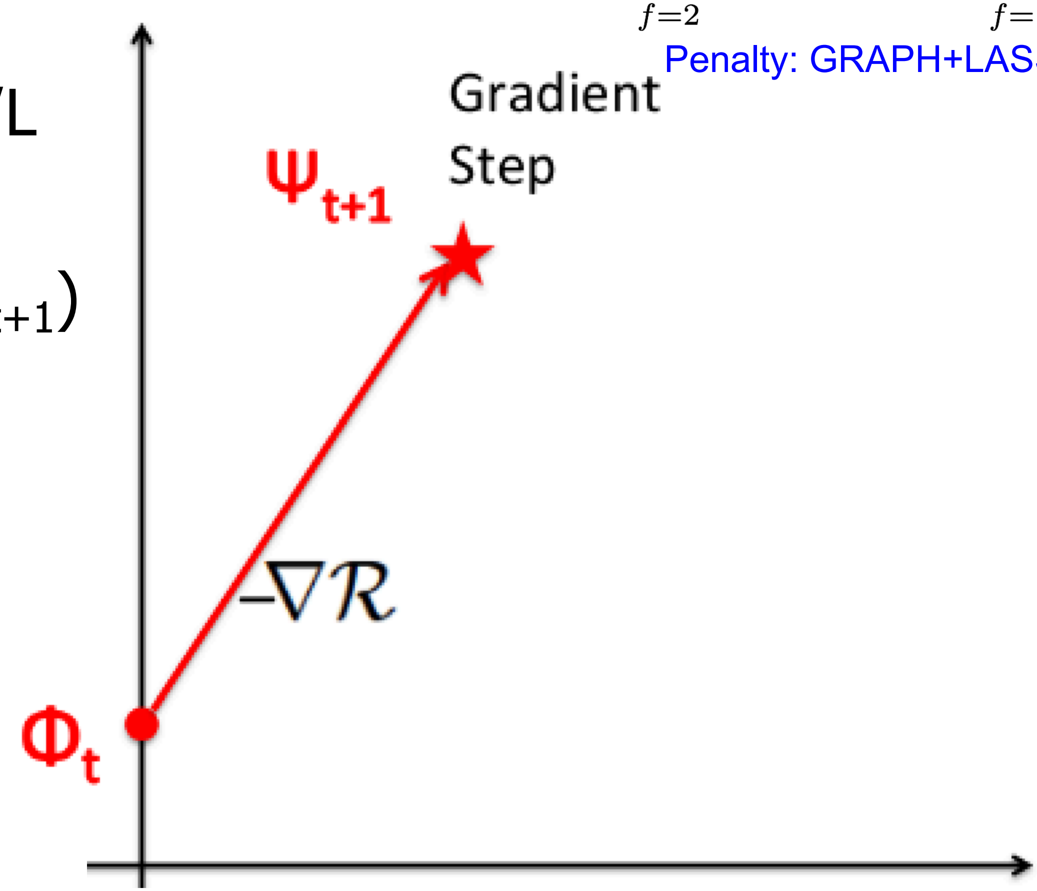
Penalty: GRAPH+LASSO

Gradient step:

$\Psi_{t+1} = \Phi_t - \nabla R(\Phi_t)/L$

Proximal step:

$\Phi_{t+1} = \text{prox}_{\Omega,L/2}(\Psi_{t+1})$



Different variables names, but that's just inconsequential cosmetics:

$v = \Psi$

$w = \Phi$

$n = t$

Proximal gradient method

Problem: minimize convex $R(\Phi) + \Omega(\Phi)$

Proximal operator: $\text{prox}_{\Omega,b}(\Psi) = \text{argmin}_{\Phi}\, \Omega(\Phi)+b||\Phi - \Psi||^2$

MM iteration:

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

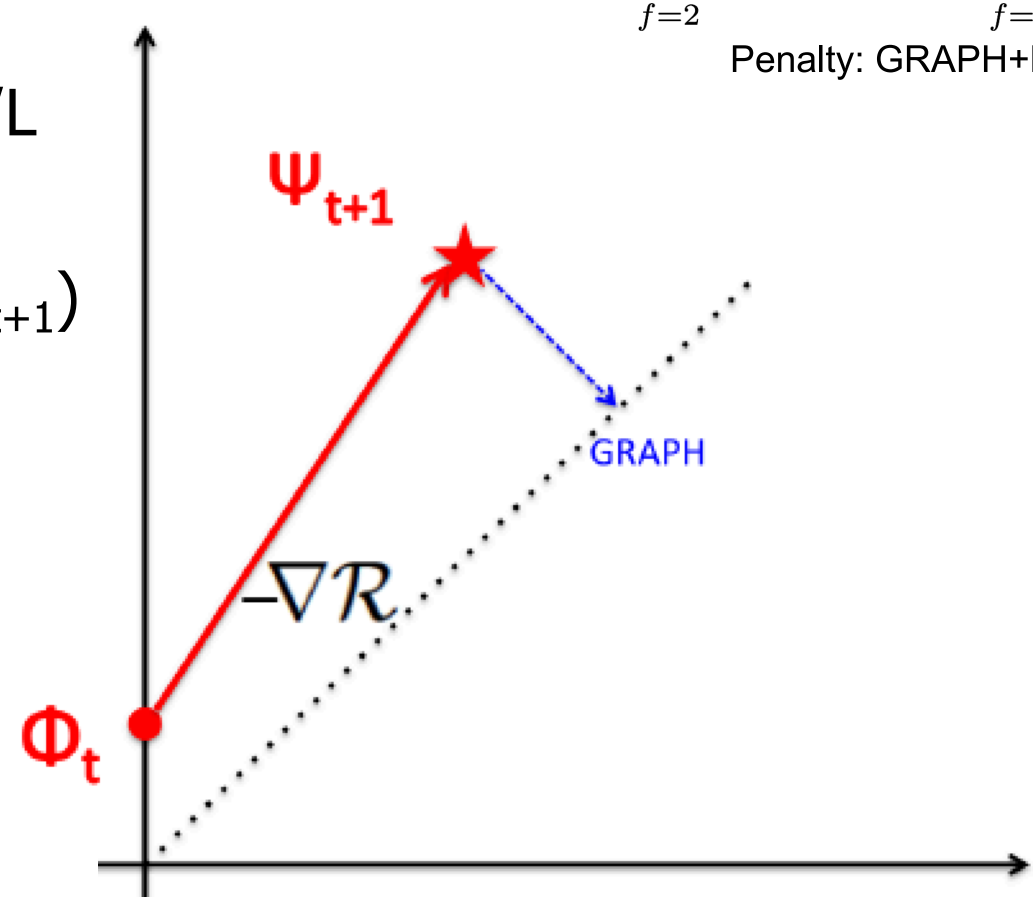
Penalty: GRAPH+LASSO

Gradient step:

$\Psi_{t+1} = \Phi_t - \nabla R(\Phi_t)/L$

Proximal step:

$\Phi_{t+1} = \text{prox}_{\Omega,L/2}(\Psi_{t+1})$



Proximal gradient method

Problem: minimize convex $R(\Phi) + \Omega(\Phi)$

Proximal operator: $\mathrm{prox}_{\Omega,b}(\Psi) = \mathrm{argmin}_{\Phi}\, \Omega(\Phi)+b||\Phi - \Psi||^2$

MM iteration:

Gradient step:

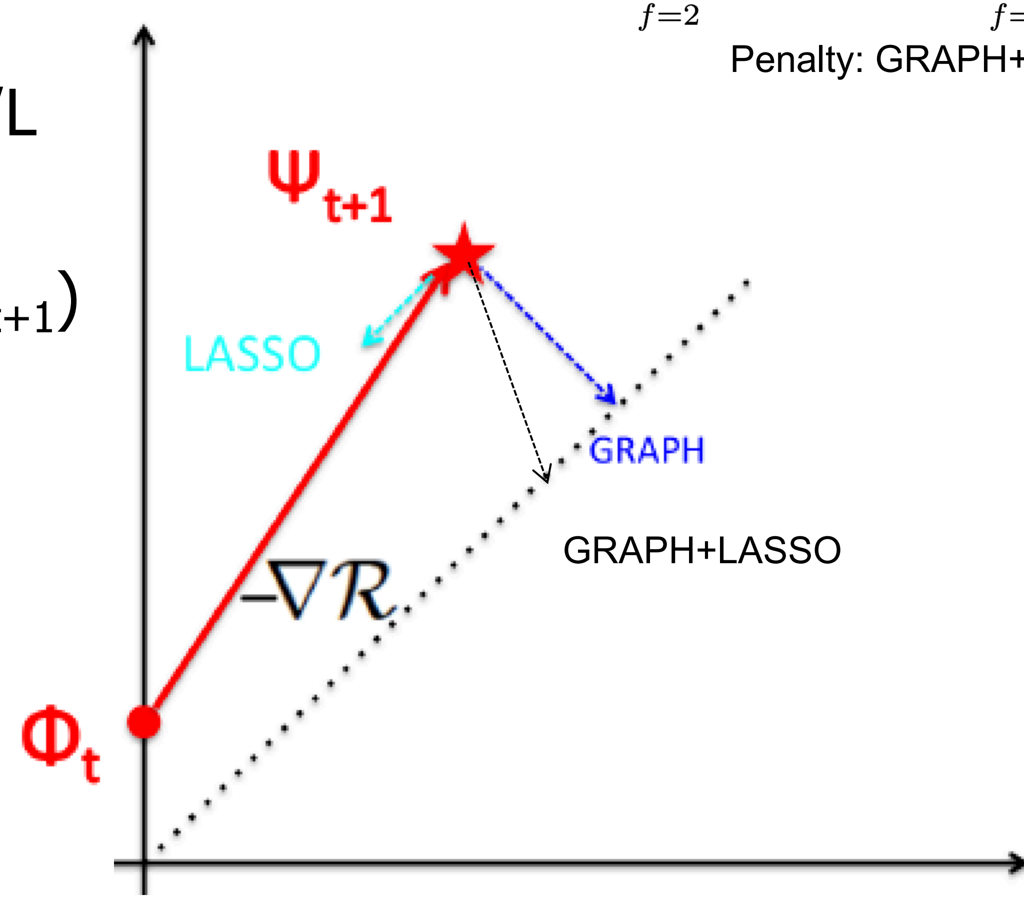
$$\Psi_{t+1} = \Phi_t - \nabla R(\Phi_t)/L$$

Proximal step:

$$\Phi_{t+1} = \mathrm{prox}_{\Omega,L/2}(\Psi_{t+1})$$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

Penalty: GRAPH+LASSO



Proximal gradient method

Problem: minimize convex $R(\Phi) + \Omega(\Phi)$

Proximal operator: $\mathrm{prox}_{\Omega,b}(\Psi) = \mathrm{argmin}_{\Phi}\, \Omega(\Phi)+b||\Phi - \Psi||^2$

MM iteration:

Gradient step:

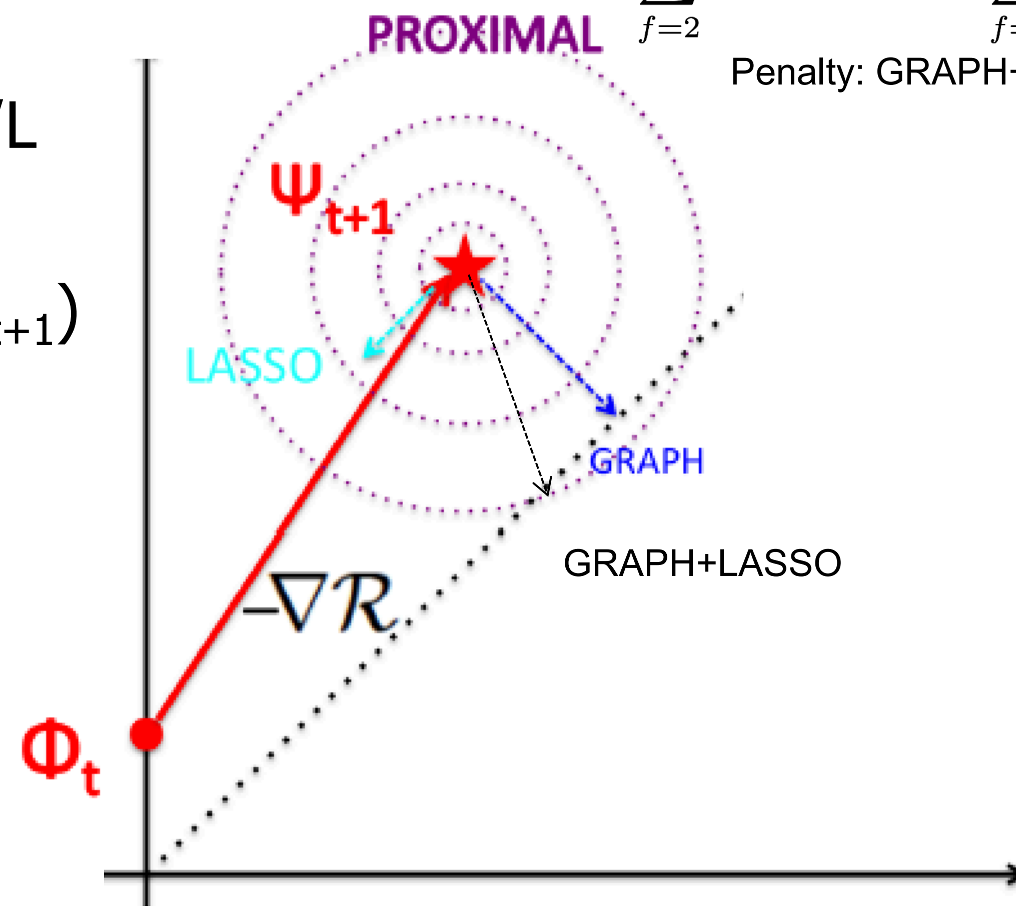
$\Psi_{t+1}= \Phi_t - \nabla R(\Phi_t)/L$

Proximal step:

$\Phi_{t+1} = \mathrm{prox}_{\Omega,L/2}(\Psi_{t+1})$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

Penalty: GRAPH+LASSO



Proximal gradient method

Problem: minimize convex $R(\Phi) + \Omega(\Phi)$

Proximal operator: $\text{prox}_{\Omega,b}(\Psi) = \text{argmin}_{\Phi}\, \Omega(\Phi)+b||\Phi - \Psi||^2$

MM iteration:

Gradient step:

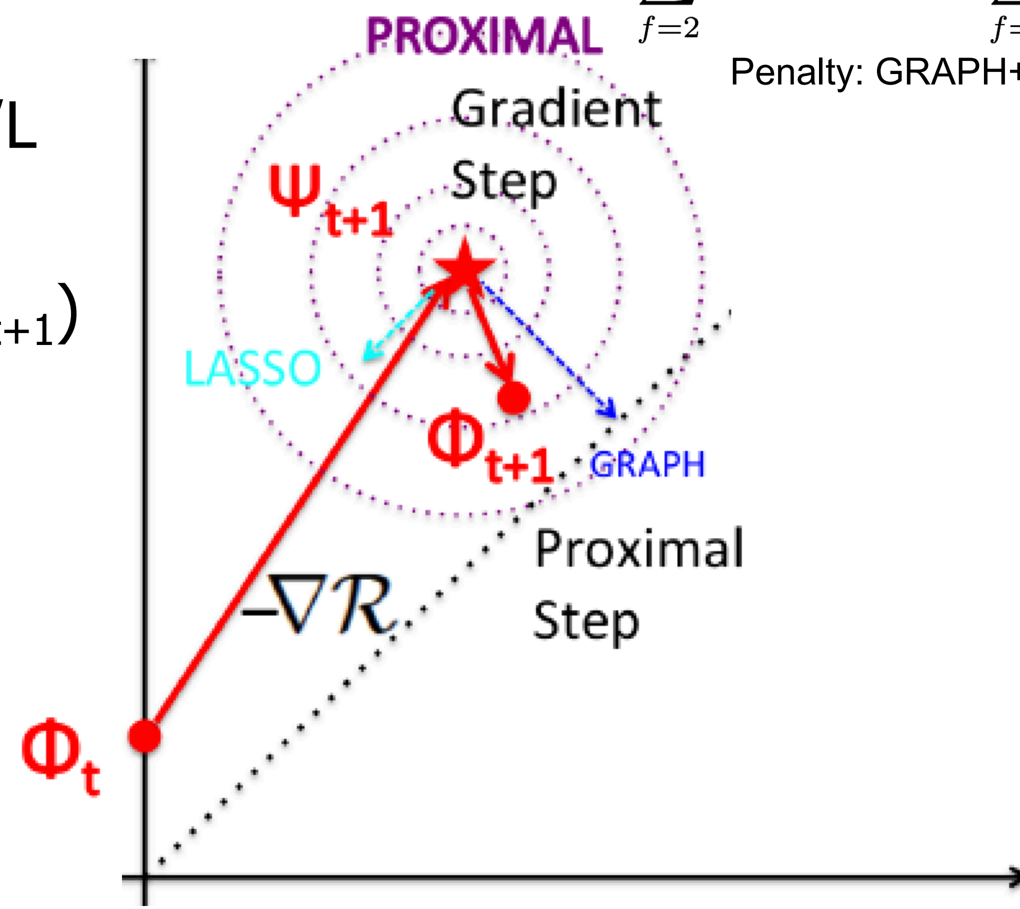
$$\Psi_{t+1} = \Phi_t - \nabla R(\Phi_t)/L$$

Proximal step:

$$\Phi_{t+1} = \text{prox}_{\Omega,L/2}(\Psi_{t+1})$$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

Penalty: GRAPH+LASSO



Ω = fused L_1 penalty + L_1 penalty

Problem: minimize regularized empirical risk: $R_S(w) + \Omega(w)$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

Proximal operator:

$\text{prox}_{\Omega,b}(v) = \text{argmin}_w \; \Sigma|w_f - w_{f-1}| + \Sigma|w_f| + b\,\Sigma\,(w_f - v_f)^2$

$\text{prox}_{\Omega,b}(v) = \text{argmin}_w \; \Sigma|w_f - w_{f-1}| + \Sigma|w_f| + b\Sigma w_f^2 + b\Sigma v_f^2 - 2b\Sigma v_f w_f$

$\text{prox}_{\Omega,b}(v) = \text{argmin}_w \; \Sigma|w_f - w_{f-1}| + \Sigma|w_f| + b\Sigma w_f^2 - 2b\Sigma v_f w_f$

Ω = fused L_1 penalty + L_1 penalty

Problem: minimize regularized empirical risk: $R_S(w) + \Omega(w)$

$$\Omega(w) = \sum_{f=2}^{F} |w_f - w_{f-1}| + \sum_{f=1}^{F} |w_f|$$

MM iterations:

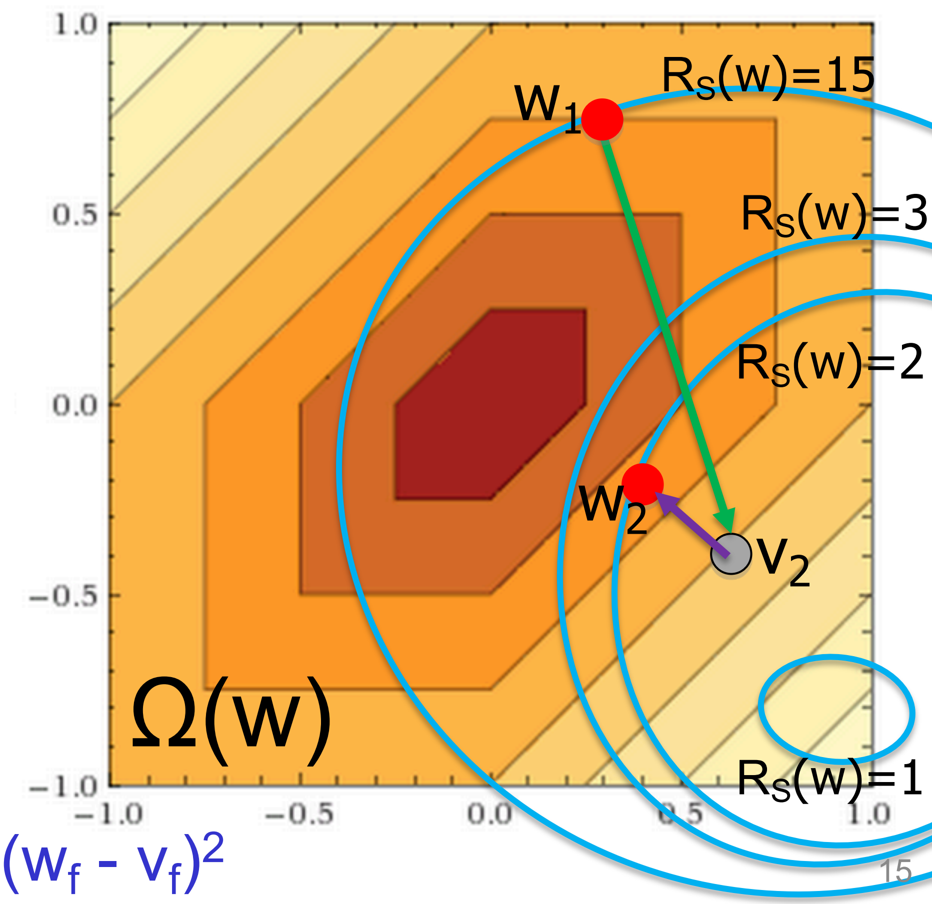
Gradient step:

$v_{n+1} = w_n - \nabla R_S(w_n)/L$

Proximal step:

$w_{n+1} = \text{prox}_{\Omega,L/2}(v_{n+1})$

Ω is not separable, so we can't treat each coordinate separately!



Proximal operator:

$\text{prox}_{\Omega,b}(v) =$

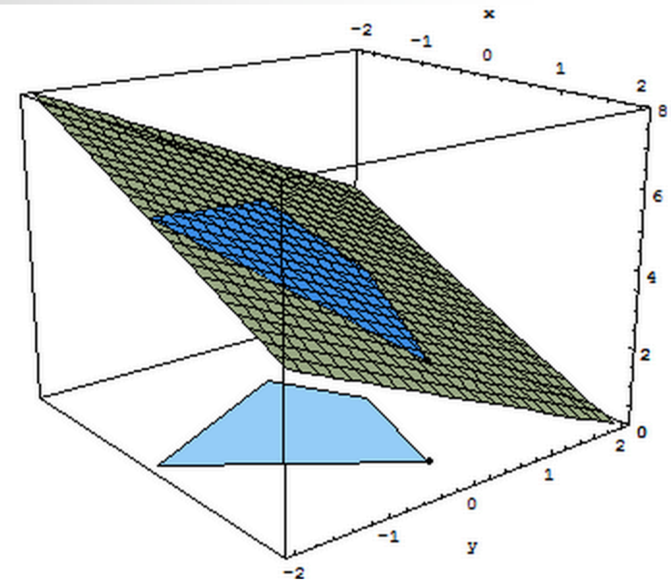
$\text{argmin}_w \; \Sigma|w_f - w_{f-1}| + \Sigma|w_f| + b\, \Sigma (w_f - v_f)^2$

Prox for fused L_1: dealing with |.|

- Shape of the objective function:
  - **Linear with linear constraints:**

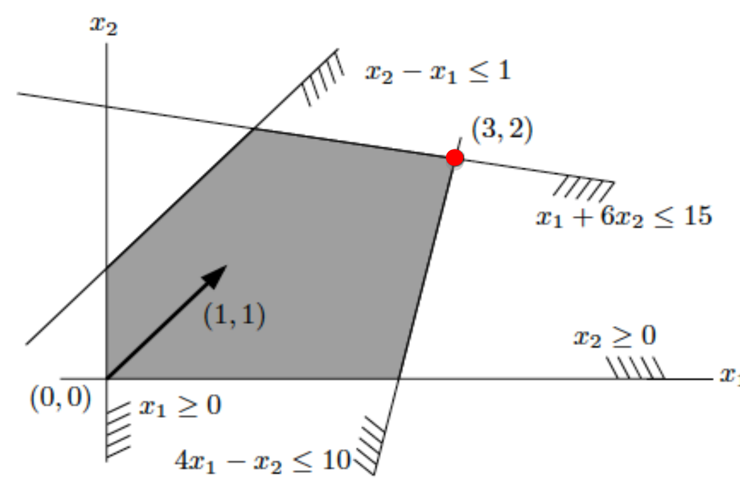
    minimize $c^T x$
    subject to: $Ax \leq b$

    x – *vector of unknown real numbers*



  - Efficiently solvable:
    - Black box: just use CPLEX/Gurobi
  - Tricks: objective function can contain terms: $c_i \max(a_i x_i, b_i x_i)$
    - as long as $c_i > 0$, we can add more constraints and transform it into:

minimize:	$c_i z_i$
subject to:	$a_i x_i \leq z_i$
	$b_i x_i \leq z_i$



    - we can deal with absolute values this way: $c_i |x_i| = c_i \max(x_i, -x_i)$

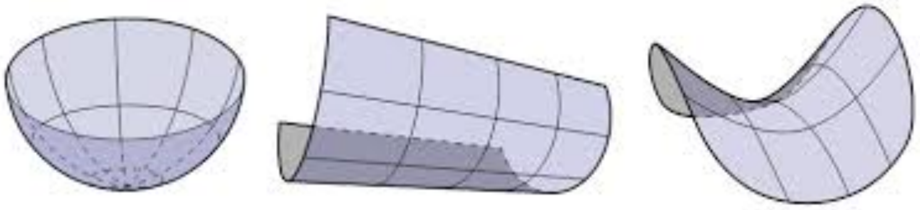
Prox for fused L_1: dealing with $(.)^2$

- Shape of the objective function: $$5x^2+8xy+5y^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
  - **Quadratic (and convex) with linear constraints:**

    minimize x^TQx+c^Tx
    subject to: $Ax \leq b$

    Q – *symmetric and positive definite matrix (thus convex shape)*

    x – *vector of unknown real numbers*



  - Efficiently solvable:
    - Standard packages: CPLEX, Gurobi
      - Black box: we don't need to care about the details
    - Specialized solvers for a given QP problem may be faster then black box
  - Problems:
    - Q should not be badly conditioned

QP: $\Sigma|w_f - w_{f-1}| + \Sigma|w_f| + b\Sigma w_f^2 - 2b\Sigma v_f w_f$

$$\text{minimize} \;\; \lambda_1 \sum_{f=2}^{F} |w_f - w_{f-1}| + \lambda_2 \sum_{f=1}^{F} |w_f| + w^T(\frac{L}{2}I)w + (-Lv)^T w$$

Expanded vector of variables $\omega = [w \; e \; z]$, new constraints:

$$\text{minimize} \;\; \omega^T Q \omega + c^T \omega$$

$$\text{subject to} \;\; A\omega \le b \;\; \begin{cases} w_f - w_{f-1} \le e_f \\ -w_f + w_{f-1} \le e_f \end{cases} \forall f \in \{2, ..., F\}$$

$$\begin{cases} w_f \le z_f \\ -w_f \le z_f \end{cases} \forall f \in \{1, ..., F\}$$

$$c = [-Lv \;\; \lambda_1 \;\; \lambda_2]$$

$$Q = \begin{bmatrix} \frac{L}{2}I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

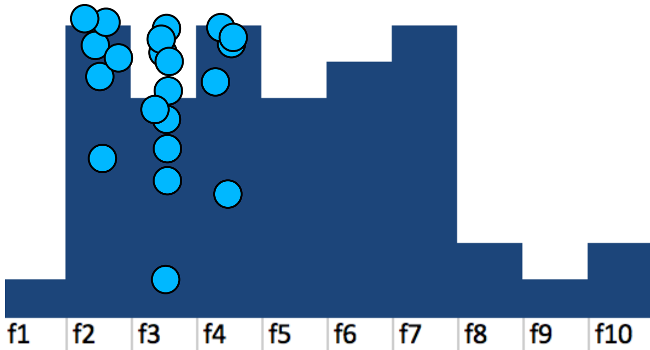
Q is positive semidefinite

We end up with a standard QP, **w** is part of solution **ω**

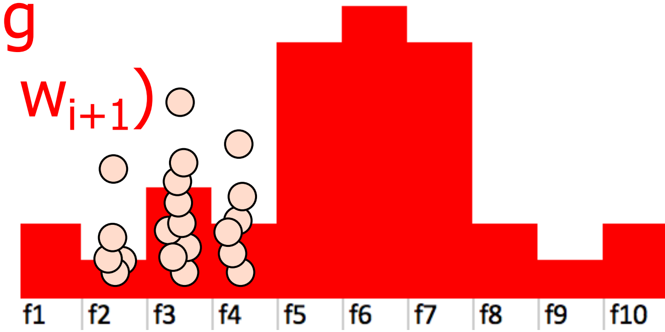
$$\text{minimize} \;\; \omega^T Q \omega + c^T \omega$$

$$\text{subject to} \;\; A\omega \le b$$

Fused Lasso – view involving sets



- Desired penalty structure:
  - If f_i is important for classification, f_{i-1} and f_{i+1} likely to be important, too
  - What we would like to have is a classifier where either both neighboring features are selected (non-zero w_i and w_{i+1}) or both are not selected ($w_i = w_{i+1} = 0$)
    - $[w_i] = [w_{i+1}]$



  - Support of a real-valued variable x is:

$$[x] = \text{supp}(x) = |\text{sign}(x)|$$

    - Support of x is 1 if x is non-zero, and is 0 otherwise
  - Similarly, for a vector w, support is the corresponding vector of 0's and 1's (1's for non-zero coordinates, 0 elsewhere)

$$[w] = \text{supp}(w) = ([w_1], [w_2], \ldots, [w_F])$$

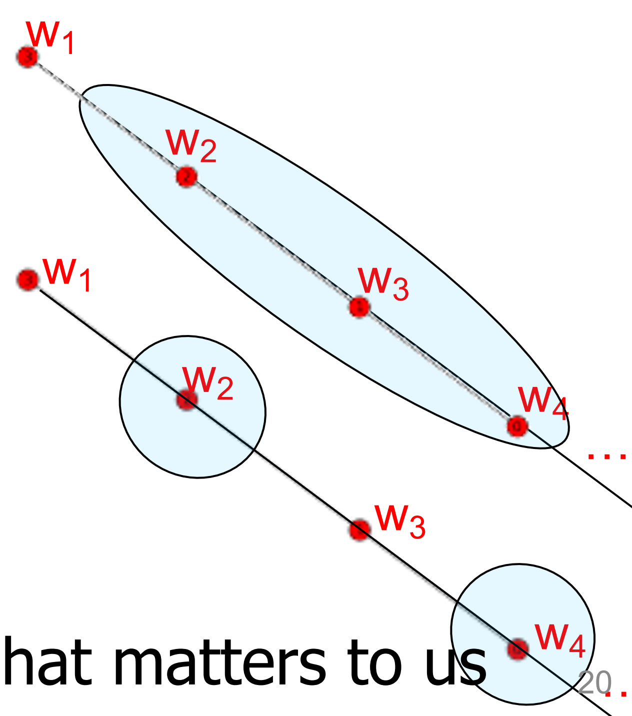
Fused Lasso – view involving sets

- If f_i is important for classification, f_{i-1} and f_{i+1} likely to be important, too
- We want a classifier where either both neighboring features are selected (non-zero w_i and w_{i+1}) or both are not selected (w_i=w_{i+1}=0):

$$\Omega(w) = \sum_{f=2}^{F} \left| [w_{f-1}] - [w_f] \right|$$

$$[x] = \mathrm{supp}(x) = |\mathrm{sign}(x)|$$

- Ω now is essentially a function defined on a set, not on a vector
  - Set of features with non-zero feature weights w_f
  - E.g. Ω($\{f_2, f_3, f_4\}$)=2 or Ω($\{f_2, f_4\}$)=4
  - Detailed values of weights are not what matters to us



Set functions

- Ω now is essentially a function defined on a set
  - Set of features with non-zero feature weights w_f
  - E.g. $\Omega(\{f_2, f_3, f_4\})=2$
- We have a large set V (the universe), Ω is defined for any subset of V, i.e., Ω: $2^V \rightarrow R$

  (2^V denotes a set of all subsets of V, including V itself, and empty set Ø)
  - V has an element corresponding to each feature, $V=\{f_1, f_2, f_3, f_4, \ldots\}$
- How is Ω defined?
  - We have a graph with V as nodes
  - For an input set S:

    $\Omega(S)$ = number of edges between S and V-S