Gradient Descent Solutions to Least-Square Problems

Objectives

- explain need for iterative algorithms
- derive gradient descent algorithm
- consider impact of step size on convergence
- introduce notion of convex functions

Iterative solution methods play an important role

Features/labels: $\underline{x}_i, d_i, i = 1, 2, \ldots N$

Classifier or model error: $e^2 = \sum_{i=1}^{N} (\underline{x}_i^T \underline{w} - d_i)^2$

$$\underline{A} = \begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix} \quad \underline{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad e^2 = \|\underline{A}\,\underline{w} - \underline{d}\|_2^2$$

Regularized least squares: $\arg\min_{\underline{w}} \|\underline{A}\,\underline{w} - \underline{d}\|_2^2 + \lambda r(\underline{w})$

1. Computational cost $(\underline{A}^T\underline{A})^{-1}$
2. Closed form solution may be unavailable
3. Adapt $\underline{w}$ to new features/labels

} develop iterative approach

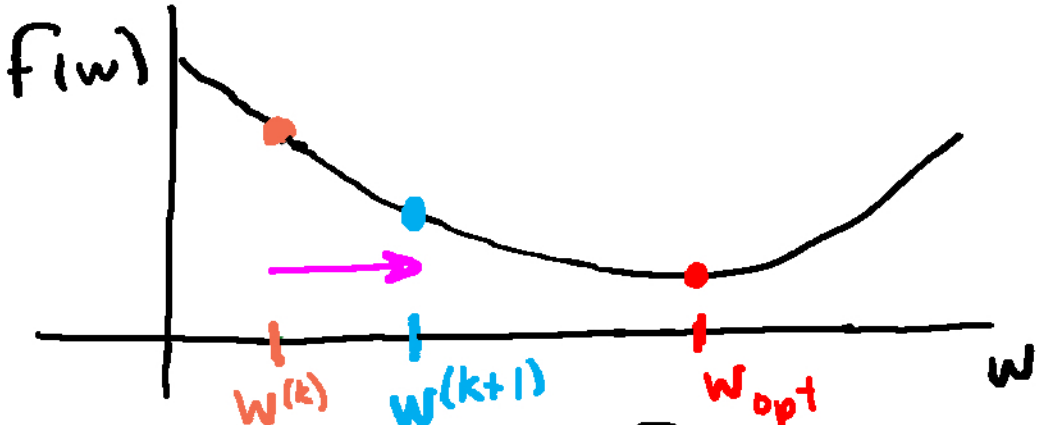
Gradient descent finds the minimum

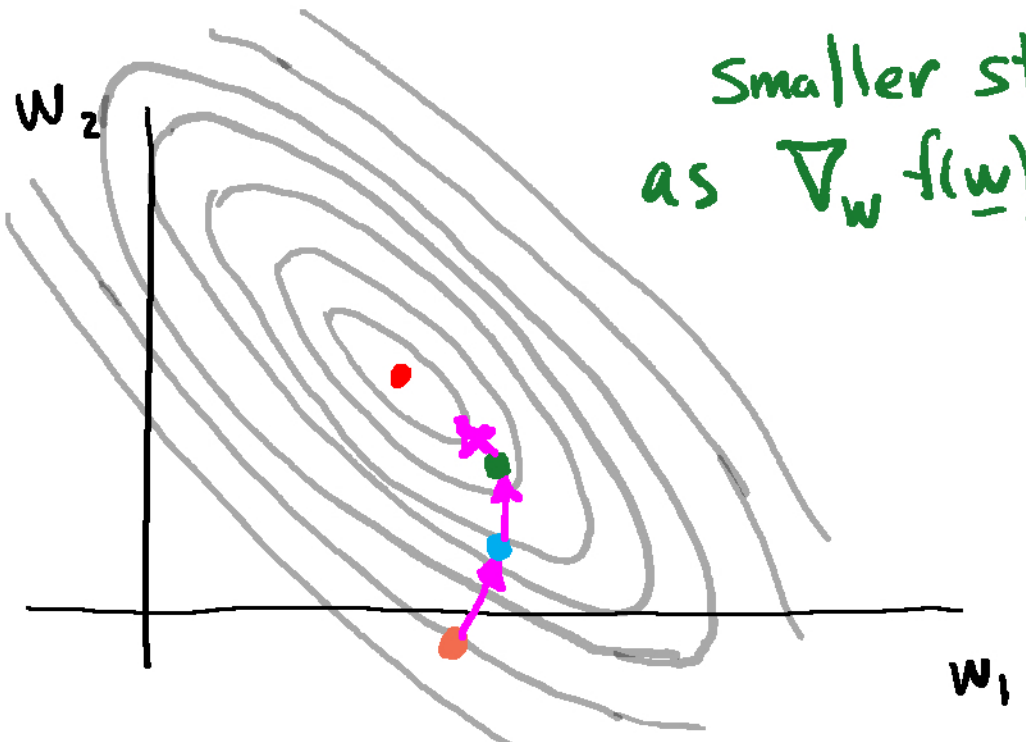
$$f(\underline{w}) = \|\underline{A}\underline{w} - \underline{d}\|_2^2$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \tau' \nabla_{\underline{w}} f(\underline{w})$$

step size; gradient

$(\tau' > 0)$





Smaller steps as $\nabla_{\underline{w}} f(\underline{w})$ ↓

$$f(\underline{w}) = (\underline{A}\underline{w} - \underline{d})^T(\underline{A}\underline{w} - \underline{d})$$

$$= \underline{w}^T\underline{A}^T\underline{A}\underline{w} - 2\underline{w}^T\underline{A}^T\underline{d} + \underline{d}^T\underline{d}$$

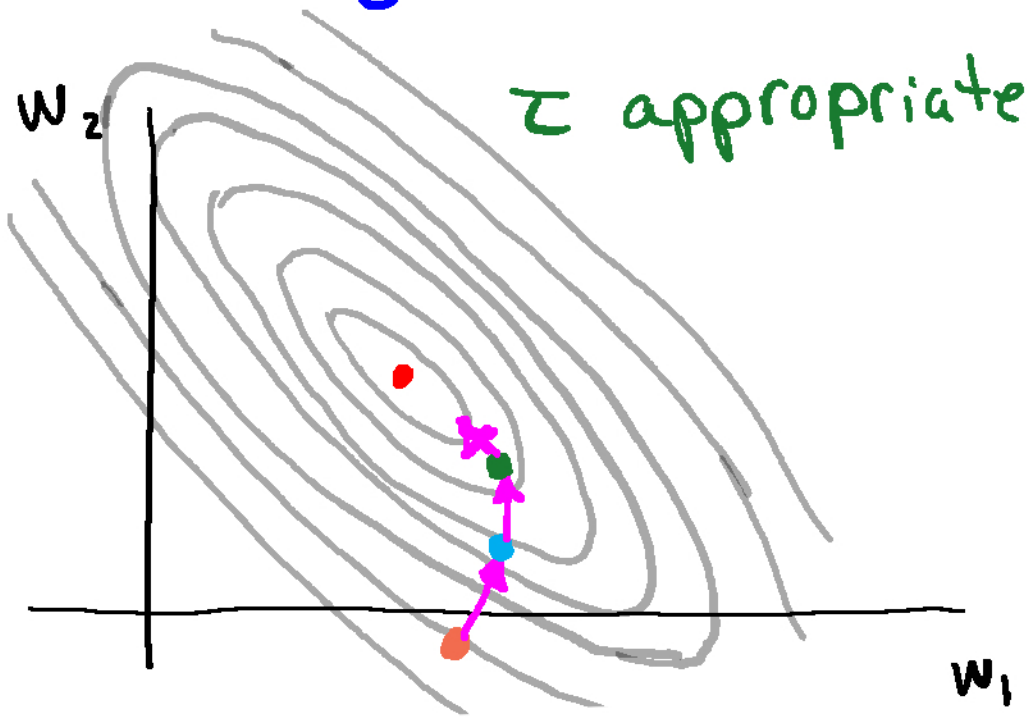
$$\nabla_{\underline{w}} f(\underline{w}) = 2\underline{A}^T\underline{A}\underline{w} - 2\underline{A}^T\underline{d}$$

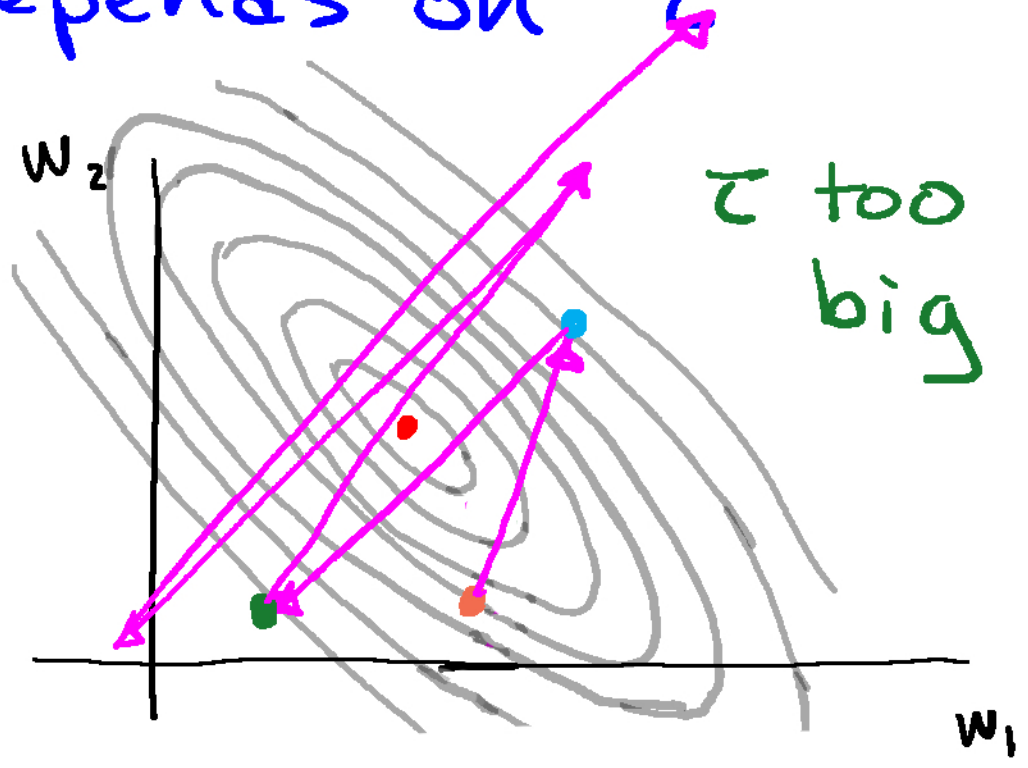
$$= 2\underline{A}^T(\underline{A}\underline{w} - \underline{d})$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \tau\,\underline{A}^T(\underline{A}\underline{w}^{(k)} - \underline{d})$$

(Landweber iteration)

Convergence behavior depends on τ





τ too small: slow convergence

τ too big: no convergence
unstable!

Require $0 < \tau < 2/\|\underline{A}\|_{op}^2$ for convergence

Recall $\|\underline{A}\|_{op} = \|\underline{A}\|_2 = \sigma_{max}(\underline{A})$

Convergence: $f(\underline{w}^{(k+1)}) < f(\underline{w}^{(k)})$ cost decreases as k increases

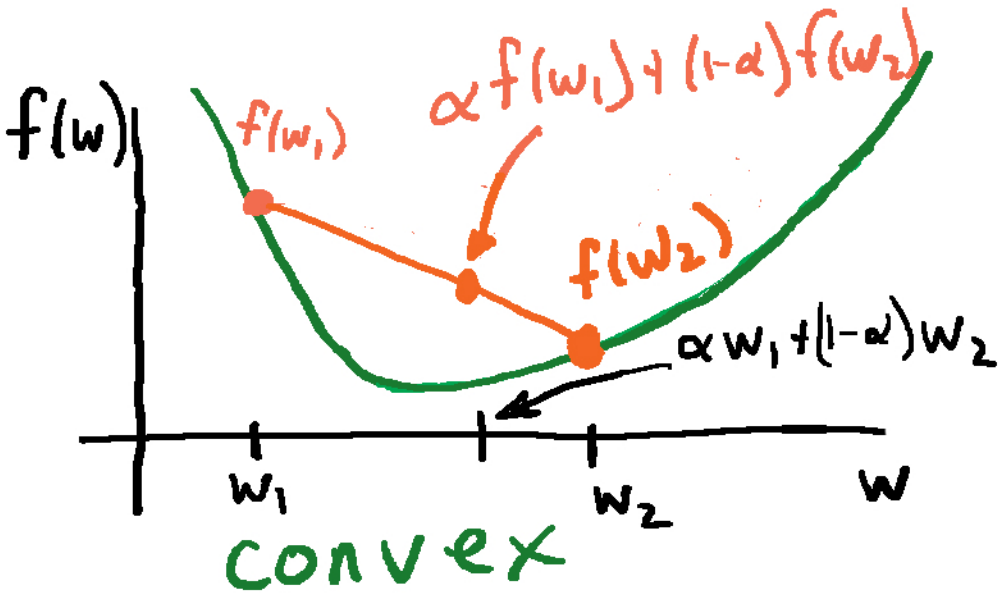
$$\|\underline{A}\underline{w}^{(k+1)} - \underline{d}\|_2^2 < \|\underline{A}\underline{w}^{(k)} - \underline{d}\|_2^2$$

Notes - guaranteed convergence for

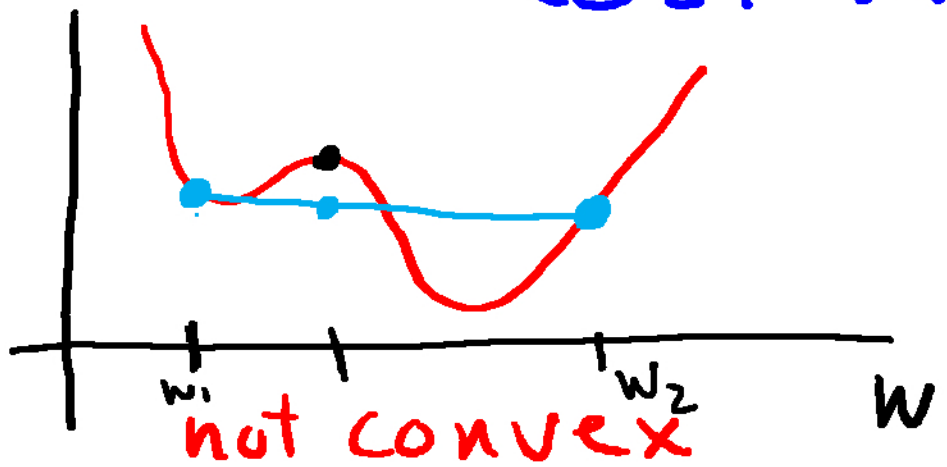
$$0 < \tau < 2/\|\underline{A}\|_{op}^2$$

$\underline{w}^{(0)} = \underline{0}$, $\underline{w}^{(k+1)} = \underline{w}^{(k)} - \tau \underline{A}^T(\underline{A}\underline{w}^{(k)} - \underline{d}) \xrightarrow{k} (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d}$

Gradient descent is effective for convex cost functions



convex



not convex

local minima

$$f(\alpha w_1 + (1-\alpha) w_2) \leq \alpha f(w_1) + (1-\alpha) f(w_2); \quad \frac{d^2}{dw^2} f(w) \geq 0$$

$0 < \alpha < 1$, all w_1, w_2

Multidimensional case

$$\underline{H}(\underline{w}) \geq 0$$

$$[\underline{H}(\underline{w})]_{ij} = \frac{\partial^2}{\partial w_i \partial w_j} f(\underline{w})$$

Proximal Gradient Descent Algorithms

Objectives

- derive proximal gradient algorithm for regularized least-squares problems
  - least-squares gradient descent
  - regularize
- apply to ridge regression

Proximal gradient descent solves regularized least-squares problems

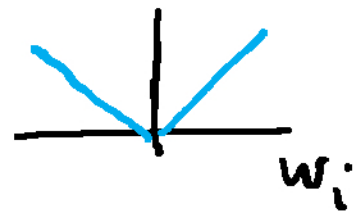
$$\min_{\underline{w}} \|\underline{A}\underline{w} - \underline{d}\|_2^2 + \lambda r(\underline{w})$$

$r(\underline{w})$: regularizer

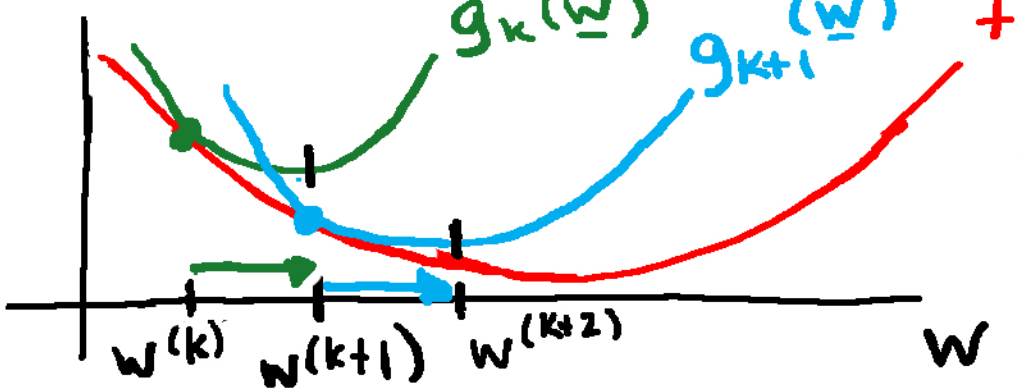
$\lambda > 0$: tuning parameter

Example Convex Regularizers

- Ridge (Tikhonov) $r(\underline{w}) = \|\underline{w}\|_2^2 = \sum_{i=1}^{m} w_i^2$
- LASSO (ℓ_1) $r(\underline{w}) = \|\underline{w}\|_1 = \sum_{i=1}^{m} |w_i|$ not differ.



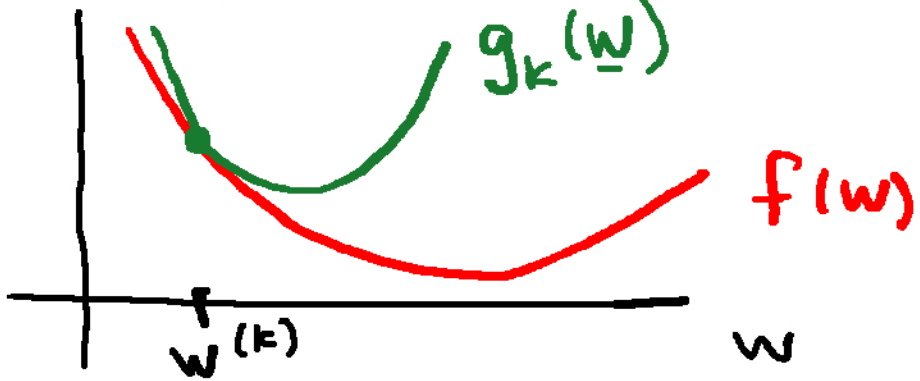
Proximal Gradient Descent Concept



$f(\underline{w}) = \|\underline{A}\underline{w} - \underline{d}\|_2^2 + \lambda r(\underline{w})$

- solve sequence of simpler problems
- simple for separable $r(\underline{w}) = \sum_i h_i(w_i)$

Find $g_k(\underline{w})$ so $f(\underline{w}) \le g_k(\underline{w})$, $g_k(\underline{w}^{(k)}) = f(\underline{w}^{(k)})$



minimize $g_k(w) \Rightarrow f(w)$ decreases

$$f(\underline{w}) = \|\underline{d} - \underline{A}\underline{w}\|_2^2 + \lambda r(\underline{w})$$

$$= \|(\underline{d} - \underline{A}\underline{w}^{(k)}) + (\underline{A}\underline{w}^{(k)} - \underline{A}\underline{w})\|_2^2 + \lambda r(\underline{w})$$

$$f(\underline{w}) = \underbrace{\|\underline{d} - \underline{A}\underline{w}^{(k)}\|_2^2}_{c_k} + \underbrace{\|\underline{A}(\underline{w}^{(k)} - \underline{w})\|_2^2}_{\le \|\underline{A}\|_{op}^2 \|\underline{w}^{(k)} - \underline{w}\|_2^2} + 2\underbrace{(\underline{d} - \underline{A}\underline{w}^{(k)})^T \underline{A}}_{\underline{v}_k^T}(\underline{w}^{(k)} - \underline{w}) + \lambda r(\underline{w})$$

Define step size $0 < \tau < 1/\|\underline{A}\|_{op}^2 \Rightarrow \frac{1}{\tau} > \|\underline{A}\|_{op}^2$

$$f(\underline{w}) \le g_k(\underline{w}) = c_k + \frac{1}{\tau}\|\underline{w}^{(k)} - \underline{w}\|_2^2 + 2\underline{v}_k^T(\underline{w}^{(k)} - \underline{w}) + \lambda r(\underline{w})$$

$g_k(\underline{w})$ is separable for $r(\underline{w})$ separable:

$$g_k(\underline{w}) = c_k + \sum_{i=1}^{m} q_i(w_i)$$

no $w_i w_j$ terms

Find $\underline{w}^{(k+1)} = \arg\min_{\underline{w}} g_k(\underline{w})$

$$g_k(\underline{w}) = c_k + \frac{1}{\tau}\|\underline{w}^{(k)} - \underline{w}\|_2^2 + 2\underline{v}_k^T(\underline{w}^{(k)} - \underline{w}) + \lambda r(\underline{w})$$

$$\tau g_k(\underline{w}) = \tau c_k + (\underline{w}^{(k)} - \underline{w})^T(\underline{w}^{(k)} - \underline{w}) + 2\tau \underline{v}_k^T(\underline{w}^{(k)} - \underline{w}) + \lambda\tau r(\underline{w})$$

$$= \tau c_k - \tau^2 \underline{v}_k^T \underline{v}_k + (\tau\underline{v}_k + (\underline{w}^{(k)} - \underline{w}))^T(\tau\underline{v}_k + (\underline{w}^{(k)} - \underline{w})) + \lambda\tau r(\underline{w})$$

$$\underline{z}^{(k)}$$

$$\underline{w}^{(k+1)} = \arg\min_{\underline{w}} \|\underline{z}^{(k)} - \underline{w}\|_2^2 + \lambda\tau r(\underline{w})$$

$$\underline{z}^{(k)} = \underline{w}^{(k)} + \tau \underline{v}_k$$

$$= \underline{w}^{(k)} + \tau \underline{A}^T(\underline{d} - \underline{A}\underline{w}^{(k)})$$

$$= \underline{w}^{(k)} - \tau \underline{A}^T(\underline{A}\underline{w}^{(k)} - \underline{d})$$

Least-squares gradient descent (Landweber)

Alternate LS gradient descent and regularization 5

$\underline{w}^{(0)} = \underline{0}$, $0 < \tau < \frac{1}{\|\underline{A}\|_{op}^2}$ initialize

$\underline{z}^{(k)} = \underline{w}^{(k)} - \tau \underline{A}^T(\underline{A}\underline{w}^{(k)} - \underline{d})$ LS gradient descent

$\underline{w}^{(k+1)} = \arg\min_{\underline{w}} \|\underline{z}^{(k)} - \underline{w}\|_2^2 + \lambda\tau r(\underline{w})$ regularize

if $\|\underline{w}^{(k+1)} - \underline{w}^{(k)}\| < \varepsilon$ stop check if converged

Regularization simple for $r(\underline{w})$ separable!

if $r(\underline{w}) = \sum_{i=1}^{M} h_i(w_i)$

$$\underline{w}^{(k+1)} = \arg\min_{w_i, i=1,\dots M} \sum_{i=1}^{M} \left((z_i^{(k)} - w_i)^2 + \lambda\tau h_i(w_i) \right)$$

M scalar minimizations

Example: Ridge Regression (Tikhonov)

$$f(\underline{w}) = \|\underline{d} - \underline{A}\underline{w}\|_2^2 + \lambda \|\underline{w}\|_2^2$$

LS gradient descent:

$$\underline{z}^{(k)} = \underline{w}^{(k)} - \tau \underline{A}^T (\underline{A}\underline{w}^{(k)} - \underline{d})$$

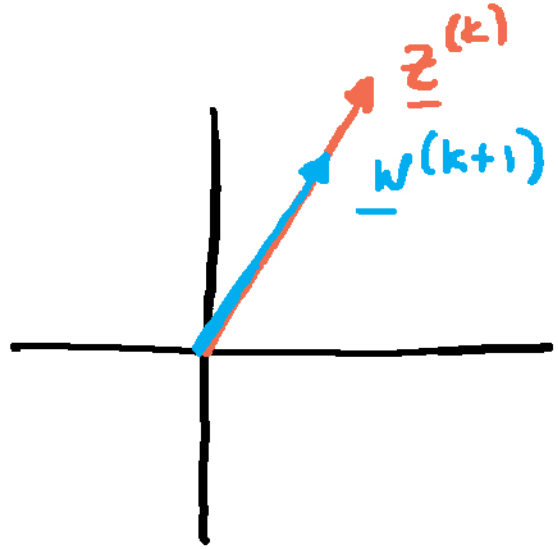
Regularization:

$$\underline{w}^{(k+1)} = \underset{w_i,\, i=1,\cdots M}{\operatorname{argmin}} \sum_{i=1}^{M} (z_i^{(k)} - w_i)^2 + \lambda\tau w_i^2$$

$$\Rightarrow w_i^{(k+1)} = \frac{1}{1+\lambda\tau} z_i^{(k)}$$

$$\underline{w}^{(k+1)} = \frac{1}{1+\lambda\tau} \underline{z}^{(k)}$$

"Shrink toward origin"



Sparse Solutions to Least-Squares Problems Using the LASSO

Objectives

- motivate search for sparse solutions
- introduce ℓ_1-norm regularization (LASSO)
- overview attributes of ℓ_1-regularization

Sparse classifiers / models give insight

$(\underline{x}_i, d_i)_{i=1,\dots N}$ features, labels

$\underline{x}_i^T \underline{w} \approx d_i$

$$\underline{A}\,\underline{w} = \begin{bmatrix} \underline{a}_1 & \underline{a}_2 & \cdots & \underline{a}_m \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \sum_{i=1}^{m} w_i \underline{a}_i$$

$\underline{a}_\ell$: ℓ^{th} feature component

Suppose $w_\ell \approx 0 \Rightarrow \underline{a}_\ell$ is unimportant

If a small number of w_ℓ are nonzero, only those few features matter! $\underline{w}$ is sparse

$$\|\underline{w}\|_0 = \sum_{i=1}^{m} \mathbb{1}_{\{w_i \neq 0\}}$$ (number of nonzero elements)

ℓ_0 "norm"

$\|a\underline{w}\|_0 \neq a\|\underline{w}\|_0$

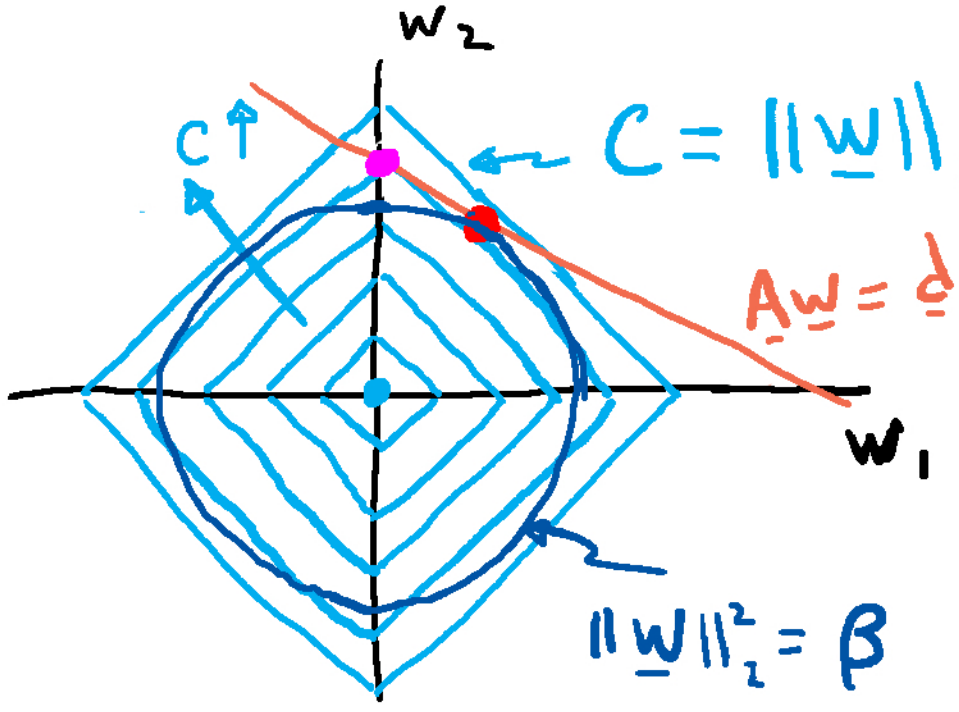
Consider $\min_{\underline{w}} \|\underline{w}\|_0$ s.t. $\|\underline{A}\underline{w} - \underline{d}\|_2^2 < \varepsilon$

non convex - intractable

Convex relaxation gives tractable problem

$\min_{\underline{w}} \|\underline{w}\|_1$ s.t. $\|\underline{A}\underline{w} - \underline{d}\|_2^2 < \varepsilon$ LASSO: Least Absolute Selection & Shrinkage Operator

convex



$c = \|\underline{w}\|_1 = \sum_{i=1}^{M} |w_i|$: $|w_1| + |w_2| = c$

$\underline{A}\underline{w} = \underline{d}$

1st quad $w_1 + w_2 = c$

$\min \|\underline{w}\|_1$ s.t. $\underline{A}\underline{w} = \underline{d}$

"corners" on $\|\underline{w}\|_1 \Rightarrow$ sparse soln's

$\min \|\underline{w}\|_2^2$ s.t. $\underline{A}\underline{w} = \underline{d}$ circular $\|w\|_2^2 \Rightarrow$ non sparse solutions

LASSO is a regularized least-squares problem 4

$$\min_{\underline{w}} \|\underline{w}\|_1 \text{ s.t. } \|\underline{A}\underline{w}-\underline{d}\|_2^2 < \varepsilon$$ is equivalent to

$$\min_{\underline{w}} \|\underline{A}\underline{w}-\underline{d}\|_2^2 + \lambda\|\underline{w}\|_1$$ for some λ, ε

Note: $\min_{\underline{w}} \|w\|_1 + \frac{1}{\lambda}\|\underline{A}\underline{w}-\underline{d}\|_2^2$

LASSO	Ridge Regression
$\underline{w}_L = \arg\min_{\underline{w}} \|\underline{A}\underline{w}-\underline{d}\|_2^2 + \lambda\|\underline{w}\|_1$	$\underline{w}_R = \arg\min_{\underline{w}} \|\underline{A}\underline{w}-\underline{d}\|_2^2 + \lambda\|\underline{w}\|_2^2$
sparse $\underline{w}_L$	non sparse $\underline{w}_R$
can have small model error $\underline{w}_{opt} - \underline{w}_L$	great prediction error $\|\underline{A}\underline{w}_{opt} - \underline{A}\underline{w}_R\|_2^2$
iterative solution	can solve in closed form

LASSO may be used for model/feature selection

$$\underline{w}_L = \arg\min_{\underline{w}} \|\underline{A}\underline{w} - \underline{d}\|_2^2 + \lambda \|\underline{w}\|_1$$

$$S_L = \{i : [\underline{w}_L]_i \neq 0\} \quad \text{selected features}$$

$$\underline{A}\underline{w}_L = \sum_{i=1}^{M} \underline{a}_i [\underline{w}_L]_i = \sum_{i \in S_L} \underline{a}_i [\underline{w}_L]_i$$

Debiasing

$$\underline{A}_L = \{\underline{a}_i : i \in S_L\}$$

$$\hat{\underline{w}}_L = \arg\min_{\underline{w}} \|\underline{A}_L \underline{w} - \underline{d}\|_2^2 = (\underline{A}_L^T \underline{A}_L)^{-1} \underline{A}_L^T \underline{d}$$

avoids shrinkage due to $\|\underline{w}\|_1$

Solving ℓ_1 Regularized Least Squares via Proximal Gradient Descent

Objectives

- apply proximal gradient approach to solve ℓ_1-regularized least squares

- derive solution to regularization phase

- explore alternating gradient and soft thresholding steps

The ℓ_1-regularized least-squares problem can be solved via proximal gradient descent

features/labels $(\underline{x}_i, d_i)$ model $\underline{x}_i^T \underline{w} \approx d_i$

$$\min_{\underline{w}} \|\underline{A}\underline{w} - \underline{d}\|_2^2 + \lambda \|\underline{w}\|_1$$

encourages sparse solutions

no closed form solution

Proximal Gradient Descent Algorithm

a) $\underline{z}^{(k)} = \underline{w}^{(k)} - \tau \underline{A}^T(\underline{A}\underline{w}^{(k)} - \underline{d})$ least squares gradient descent

b) $\underline{w}^{(k+1)} = \arg\min_{\underline{w}} \|\underline{z}^{(k)} - \underline{w}\|_2^2 + \tau\lambda\|\underline{w}\|_1$

Regularization step involves scalar minimization[3]

$$\min_{\underline{w}} \|\underline{z}^{(k)} - \underline{w}\|_2^2 + \tau\lambda\|w\|_1 \Rightarrow \min_{w_i, i=1,\cdots M} \sum_{i=1}^{M} (z_i^{(k)} - w_i)^2 + \lambda\tau|w_i|$$

Consider $\min_{w_i} (z_i^{(k)} - w_i)^2 + \lambda\tau|w_i|$, $\lambda, \tau > 0$

case 1: $w_i \geq 0$

$$\min_{w_i} (z_i - w_i)^2 + \lambda\tau w_i, \quad w_i \geq 0$$

$$\frac{d}{dw_i}\{(z_i - w_i)^2 + \lambda\tau w_i\} = 0, \quad w_i \geq 0$$

$$-2(z_i - w_i) + \lambda\tau = 0, \quad w_i \geq 0$$

$$w_i = z_i - \frac{\lambda\tau}{2}, \quad w_i \geq 0$$

if $z_i > \frac{\lambda\tau}{2}$, $w_i = z_i - \frac{\lambda\tau}{2}$

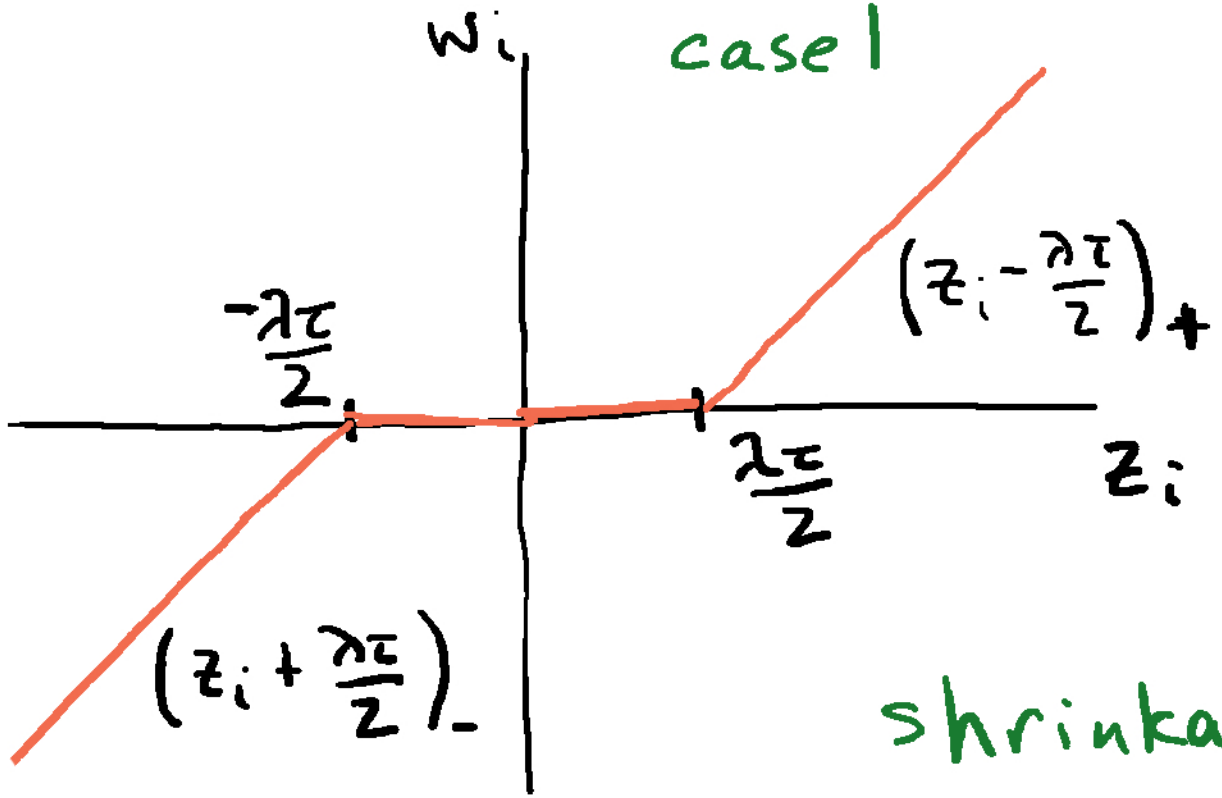
if $z_i < \frac{\lambda\tau}{2}$, $w_i = 0$

$$w_i = \left(z_i - \frac{\lambda\tau}{2}\right)_+$$

Case 2: $w_i \le 0$

$$\min_{w_i} (z_i - w_i)^2 - \lambda\tau w_i \Rightarrow \frac{d}{dw_i}\left\{(z_i - w_i)^2 - \lambda\tau w_i\right\} = 0$$

$$-2(z_i - w_i) - \lambda\tau = 0, \; w_i \le 0 \qquad w_i = \left(z_i + \frac{\lambda\tau}{2}\right)_-$$



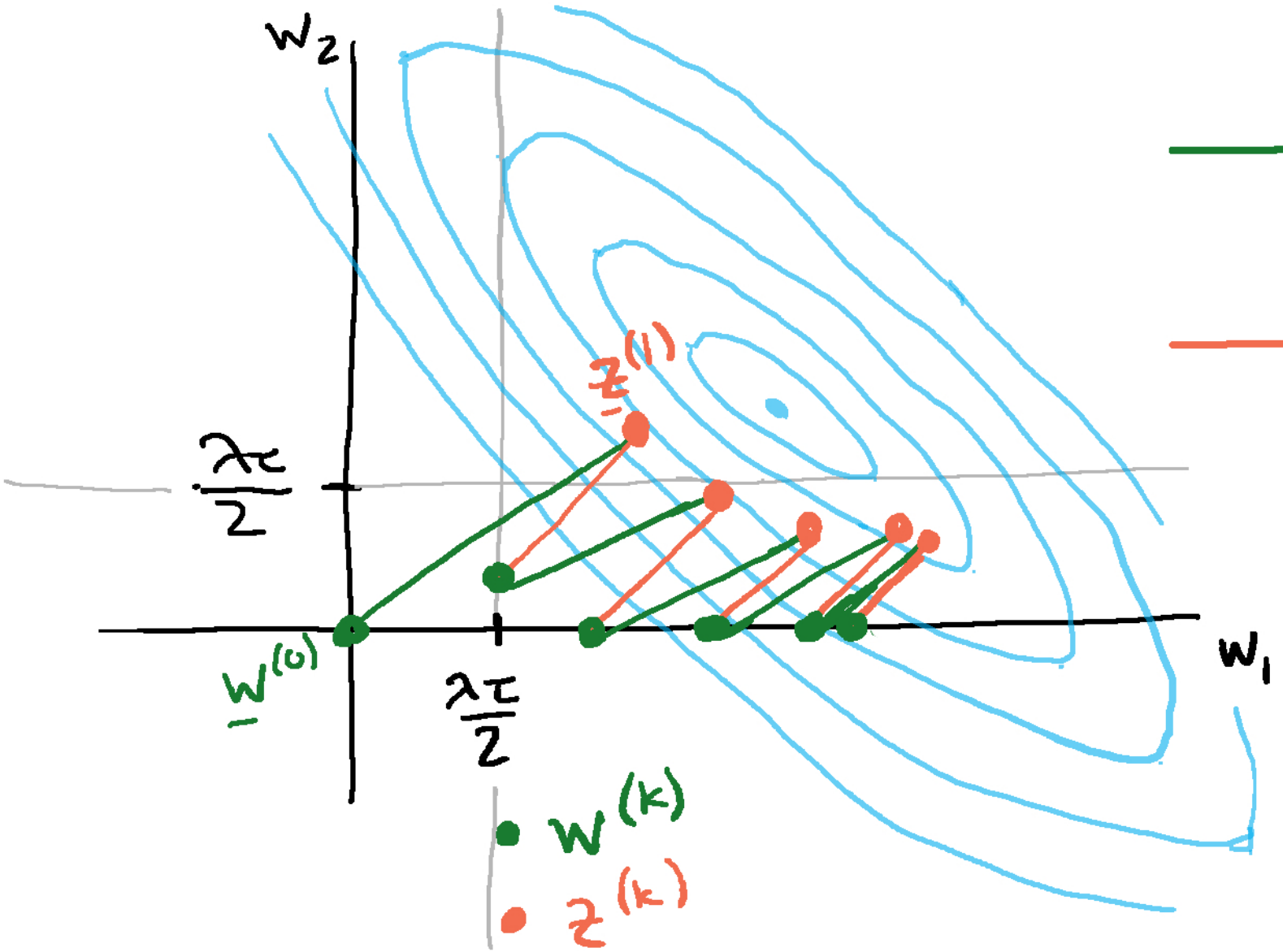
"Soft threshold"

$$w_i = \begin{cases} 0, & -\frac{\lambda\tau}{2} < z_i < \frac{\lambda\tau}{2} \\ z_i - \frac{\lambda\tau}{2}, & z_i > \frac{\lambda\tau}{2} \\ z_i + \frac{\lambda\tau}{2}, & z_i < -\frac{\lambda\tau}{2} \end{cases}$$

shrinkage

$$w_i = \left(|z_i| - \frac{\lambda\tau}{2}\right)_+ \operatorname{sign}(z_i)$$

Algorithm alternates descent and shrinkage



gradient descent

regularization (shrinkage)

Hinge Loss for Binary Classifiers

Objectives

- introduce disadvantage of squared error for classification

- introduce hinge loss cost function

- characteristics of hinge loss

Squared error "loss" can be problematic

Classifier design $\min_{\underline{w}} \ell(\underline{w}; \underline{A}, \underline{d}) + \lambda r(\underline{w})$ ← regularizer; ← loss function

Squared error loss $\ell(\underline{w}; \underline{A}, \underline{d}) = \|\underline{A}\underline{w} - \underline{d}\|_2^2$

Example: dwarf planet vs. planet

object		Ceres	Eris	Pluto	Mercury	Earth	Jupiter
x_i	radius ($\times 10^6$ m)	1.0	2.3	2.4	4.9	12.8	143.0
d_i	label	-1	-1	-1	1	1	1

$$\underline{A} = \begin{bmatrix} 1 & 1 \\ 2.3 & 1 \\ 2.4 & 1 \\ 4.9 & 1 \\ 12.8 & 1 \\ 143 & 1 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

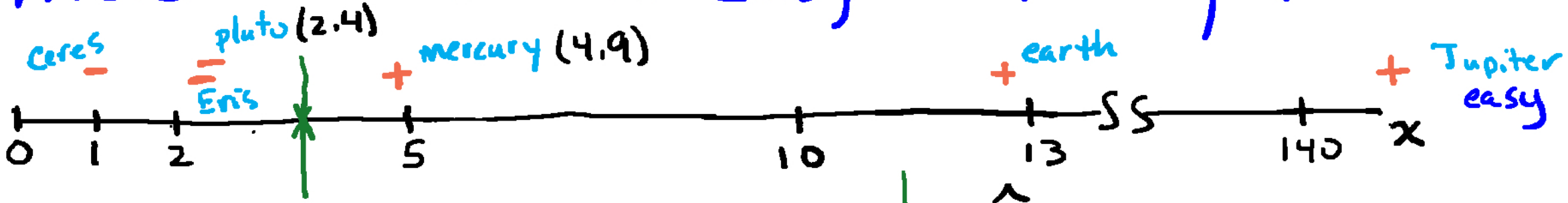
$$\underline{w}_{LS} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d} \approx 0.01 \begin{bmatrix} 1 \\ -28 \end{bmatrix}$$

dwarf: $x_i < 28$ (earth!)

planet: $x_i > 28$

squared error → poor classification

Avoid loss due to "easy-to-classify" data



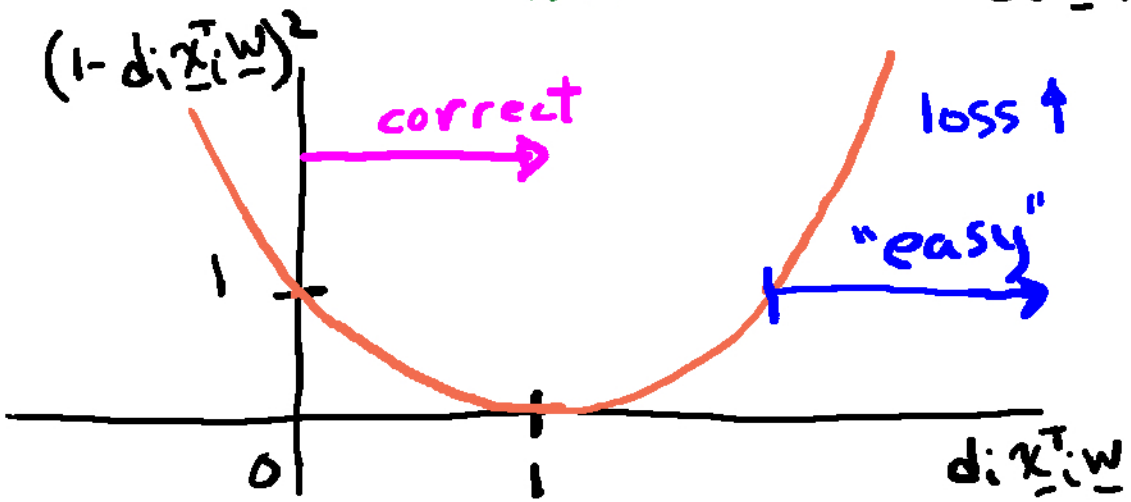
max margin classifier: midpoint $\hat{d} = \text{sign}(x - 3.65)$

Margin $4.9 - 2.4 = 2.5$ (class separation)

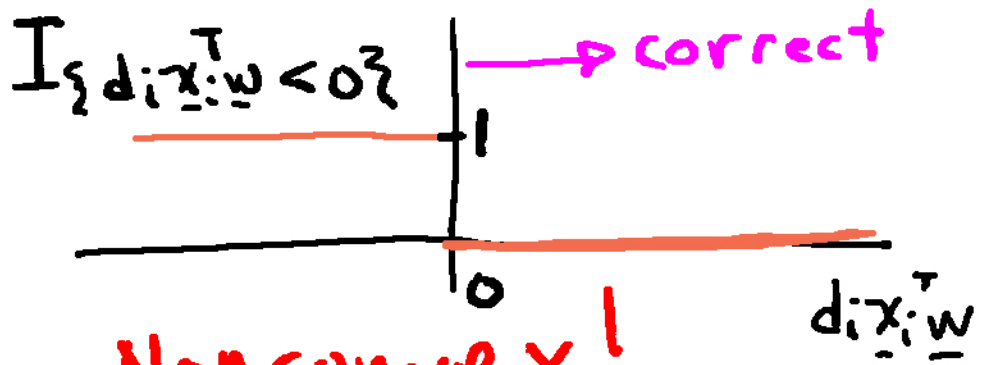
Squared error loss: $\|A\underline{w} - \underline{d}\|_2^2 = \sum_{i=1}^{N} (d_i - \underline{x}_i^T \underline{w})^2 = \sum_i (1 - d_i \underline{x}_i^T \underline{w})^2$

$(d_i = \pm 1)$

correct classification: $d_i \underline{x}_i^T \underline{w} > 0$



Ideal loss

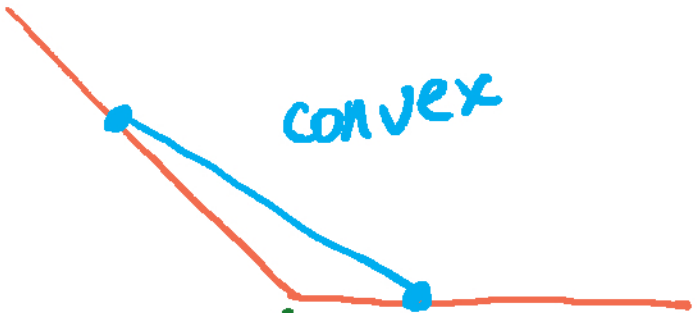


Nonconvex!

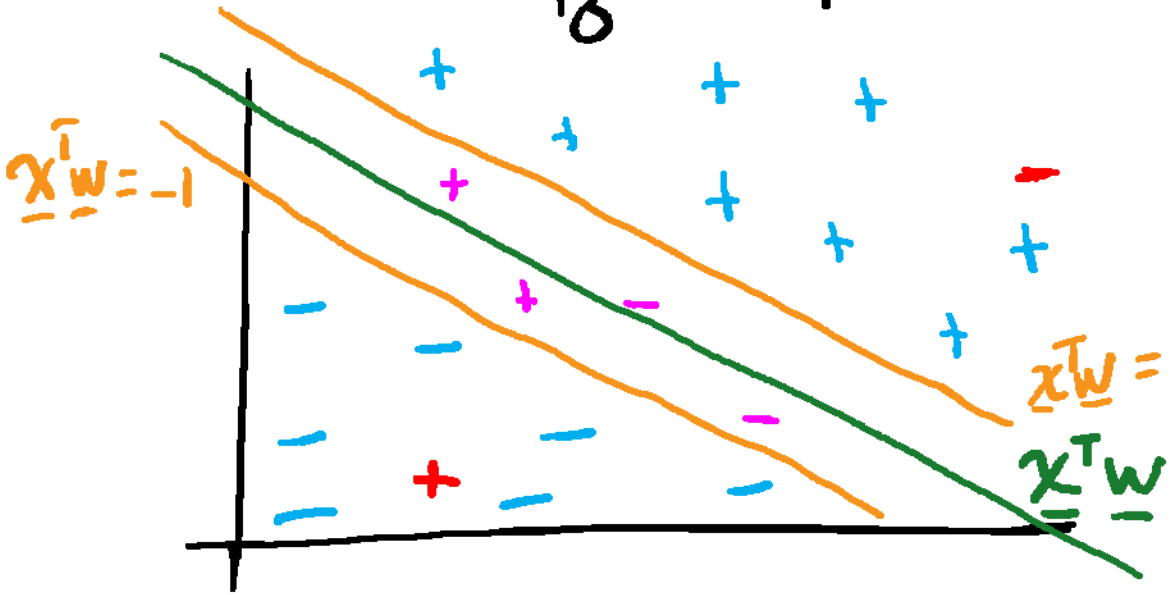
Hinge loss is convex and has no loss for easy to classify data

$$l(\underline{w}; \underline{A}, \underline{d}) = \sum_{i=1}^{N} (1 - d_i \underline{x}_i^T \underline{w})_+$$

$$(\alpha)_+ = \begin{cases} \alpha & \alpha > 0 \\ 0 & \alpha \leq 0 \end{cases}$$



not differentiable



+,− no hinge loss

+,− small hinge loss

+,− large hinge loss

Hinge loss better approximates ideal: number of misclassifications

Iterative algorithms required for finding minimum hinge loss classifier

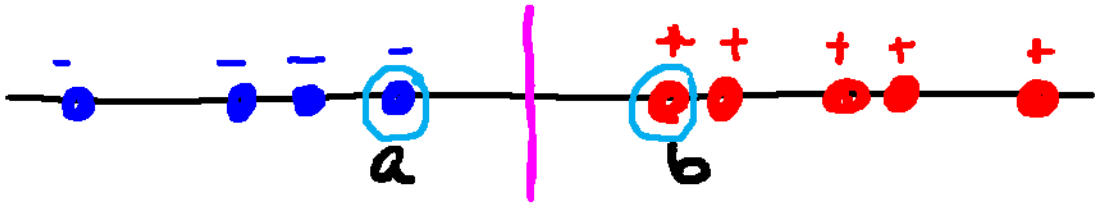
Support Vector Machines for Classification

Objectives

- Define margin for separable data
- Show support vector machines
  Maximize margin
- Use hinge loss to define support vector machines for non separable data

Maximize margin for separable training data

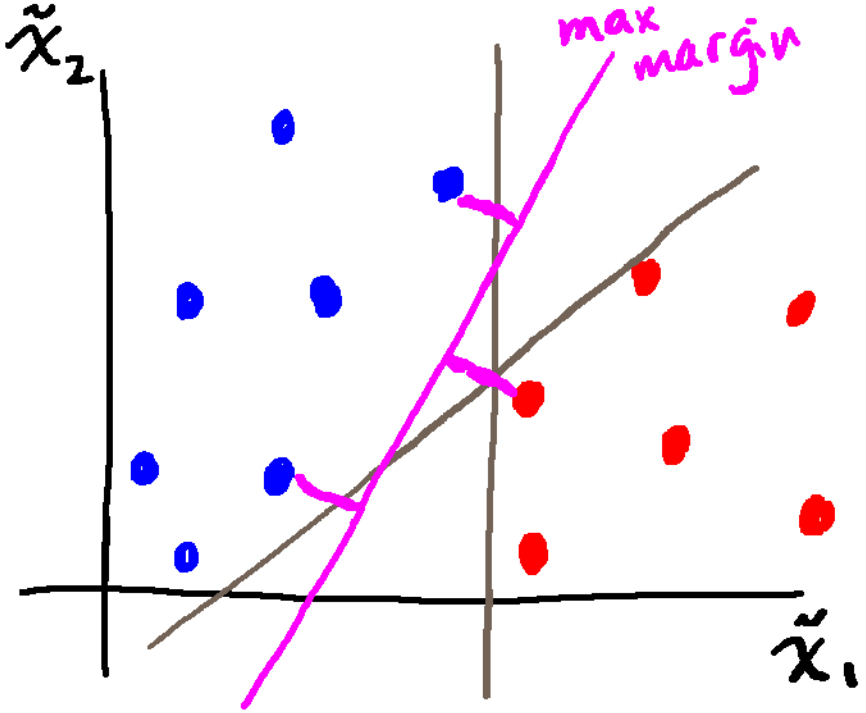
Example:



decision boundary?

margin: distance from boundary to nearest sample

max margin boundary: midpoint
only a, b matter



decision boundary?

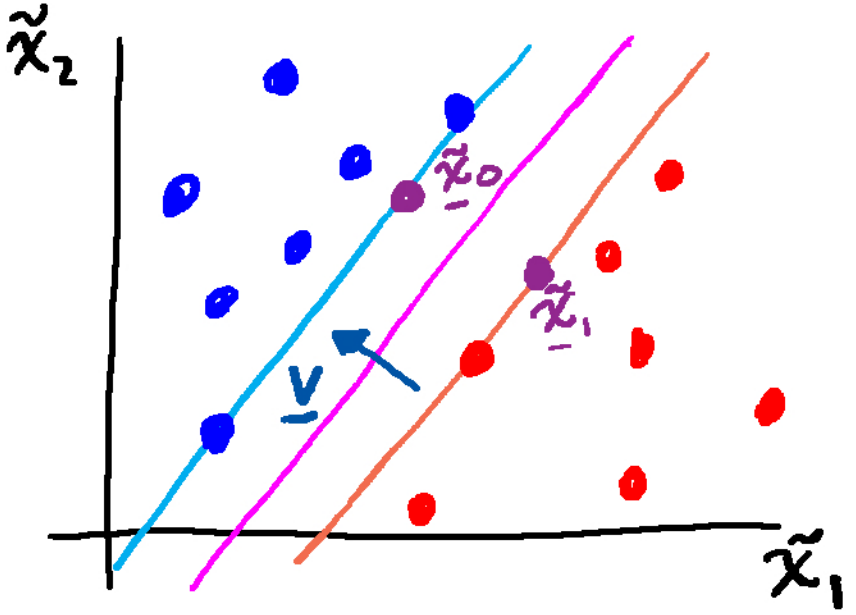
feature $\underline{x}^T = [\underline{\tilde{x}}^T \quad 1]$

weights $\underline{w}^T = [\underline{\tilde{w}}^T \; w_0]$

decision $\hat{d} = \text{sign}(\underline{x}^T \underline{w})$

$$\hat{d} = \begin{cases} 1 & \underline{\tilde{x}}^T \underline{\tilde{w}} + w_0 > 0 \\ -1 & \underline{\tilde{x}}^T \underline{\tilde{w}} + w_0 < 0 \end{cases}$$

Margin is determined by $\|\tilde{w}\|_2^{-1}$



label "-1": $\tilde{\underline{x}}^T \tilde{\underline{w}} + w_0 \leq -1$

label "+1": $\tilde{\underline{x}}^T \tilde{\underline{w}} + w_0 \geq 1$

boundary: $\tilde{\underline{x}}^T \tilde{\underline{w}} + w_0 = 0$

margin: ½ distance between // //
measure in direction $\underline{v}$

Unit normal to boundary plane: $\underline{v} = \frac{\tilde{\underline{w}}}{\|\tilde{\underline{w}}\|_2}$

Margin $m = \frac{1}{2}\|\tilde{\underline{x}}_1 - \tilde{\underline{x}}_0\|_2$ $\quad \tilde{\underline{x}}_1 = \tilde{\underline{x}}_0 + 2m\,\underline{v}$

but $\tilde{\underline{x}}_0^T \tilde{\underline{w}} + w_0 = -1$

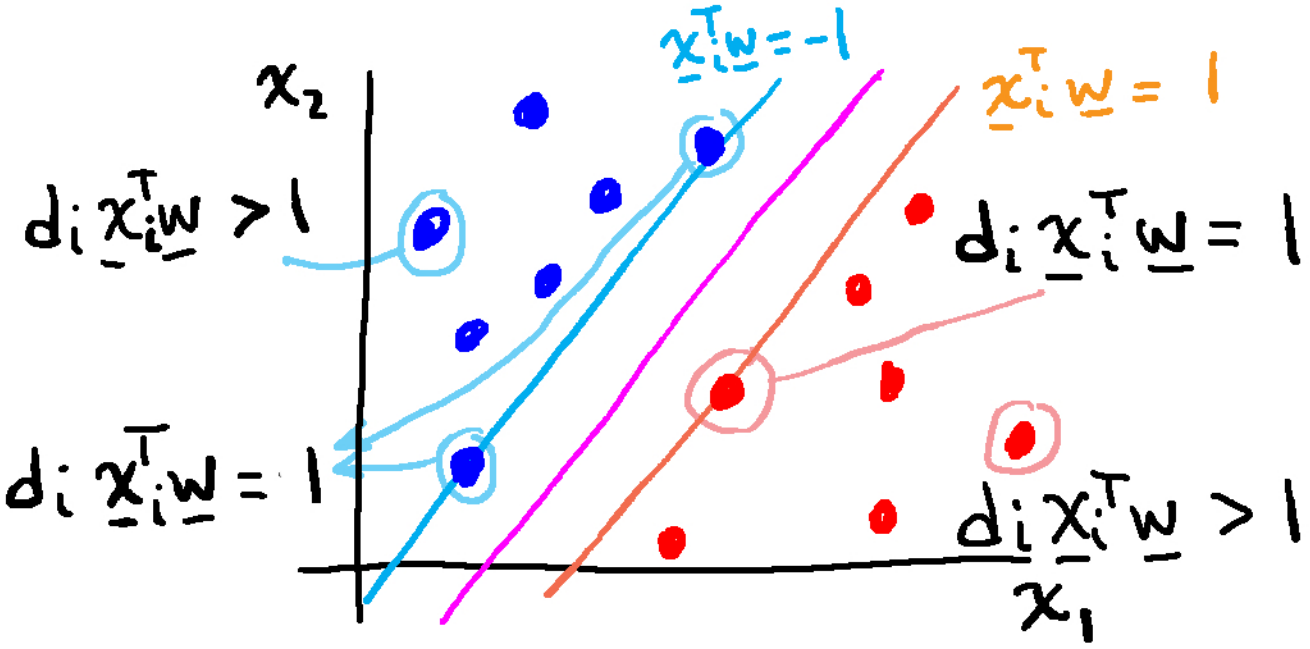
so $2 = 2m\|\tilde{\underline{w}}\|_2$

$$1 = \tilde{\underline{x}}_1^T \tilde{\underline{w}} + w_0 = \tilde{\underline{x}}_0^T \tilde{\underline{w}} + 2m \frac{\tilde{\underline{w}}^T}{\|\tilde{\underline{w}}\|_2} \tilde{\underline{w}} + w_0$$

$$\underline{m = \|\tilde{\underline{w}}\|_2^{-1}}$$

Support Vector Machine maximizes margin 4

Correct classification: $d_i(\tilde{x}_i^T \tilde{w} + w_0) \geq 1$



SVM:

$$\min_{\tilde{w}, w_0} \|\tilde{w}\|_2^2 \quad \text{s.t.} \quad d_i(\tilde{x}_i^T \tilde{w} + w_0) \geq 1, \; i = 1, 2, \ldots, N$$

$\min_{\tilde{w}, w_0} \|\tilde{w}\|_2^2$: max margin

$d_i(\tilde{x}_i^T \tilde{w} + w_0) \geq 1$: perfect classification

unique solution

Boundary defined by x_i for which $d_i x_i^T w = 1$

Called Support Vectors

SVM for non separable data uses hinge loss

$$\min_{\underline{w}} \sum_{i=1}^{N} (1 - d_i \underline{x}_i^T \underline{w})_+ + \lambda \|\underline{\tilde{w}}\|_2^2$$

loss

$\sim (\text{margin})^{-2}$

(ℓ_2 regularization)

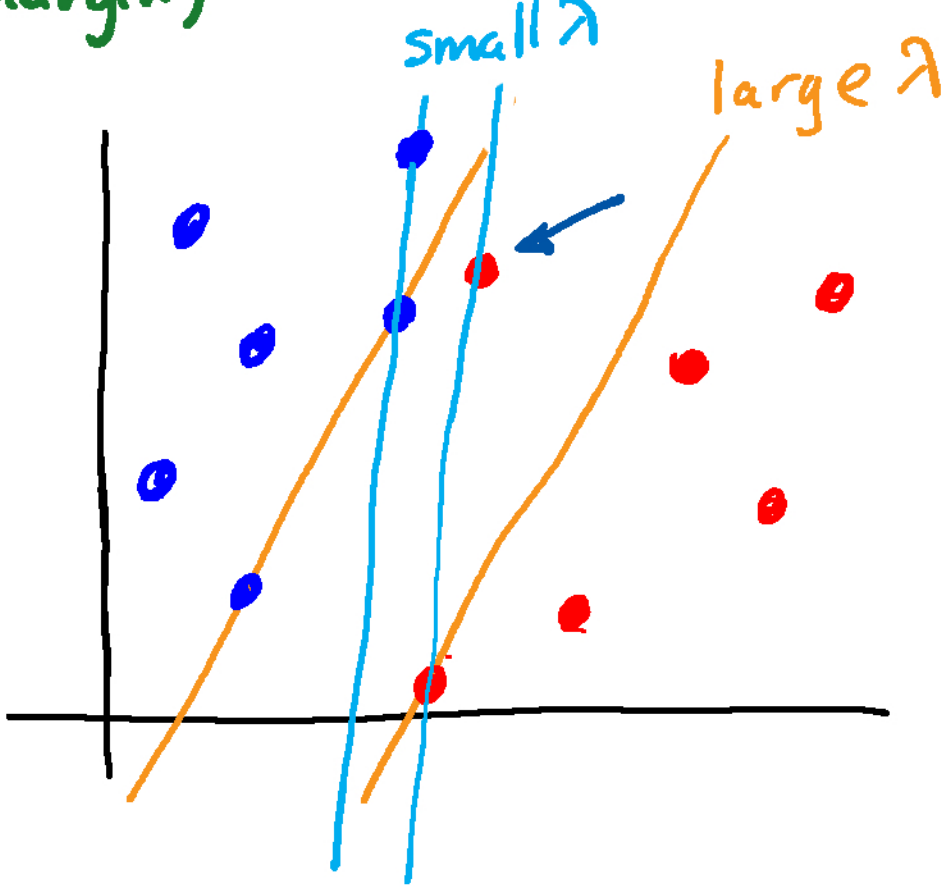
misclassification loss

non zero loss

support vectors

m

$m \sim \frac{1}{\|\underline{\tilde{w}}\|_2}$



Gradient Descent for Support Vector Machines and Subgradients

Objectives

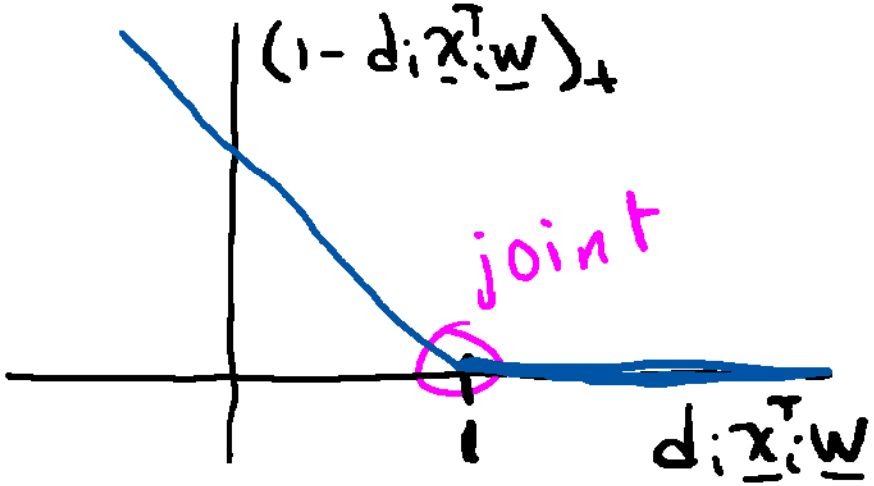
- develop a gradient descent algorithm for SVMs

- introduce subgradients for convex but non differentiable cost functions

Support vector machines require iterative algorithms

$$\min_{\underline{w}} \sum_{i=1}^{N} (1 - d_i \underline{x}_i^T \underline{w})_+ + \lambda \|\underline{w}\|_2^2$$

labels: d_i — features: $\underline{x}_i$ — hinge loss: $(\cdot)_+$ — regularization: $\lambda \|\underline{w}\|_2^2$



No closed form solution

Convex

$\Rightarrow$ gradient descent

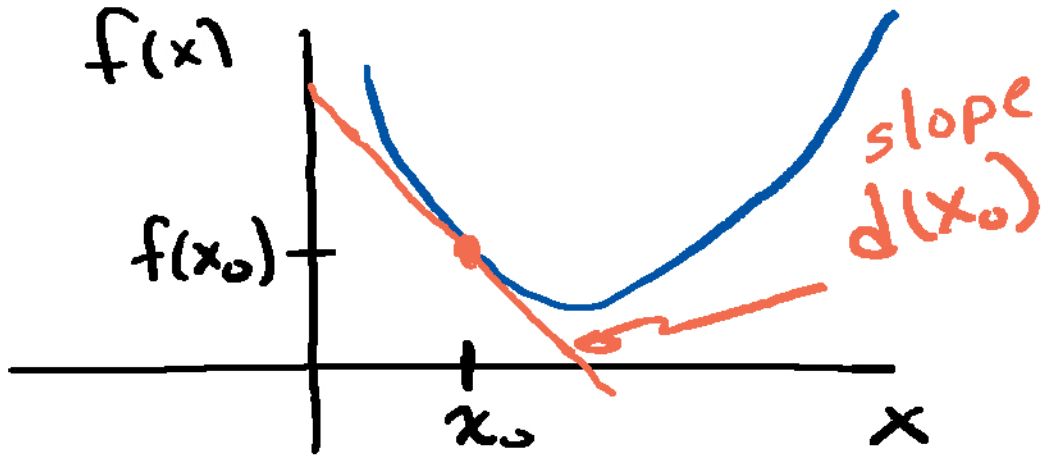
Problem: hinge loss not differentiable

Subderivatives generalize derivatives

– Convex, but nondifferentiable $f(x)$

Derivatives –

$$d(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$



Convex:

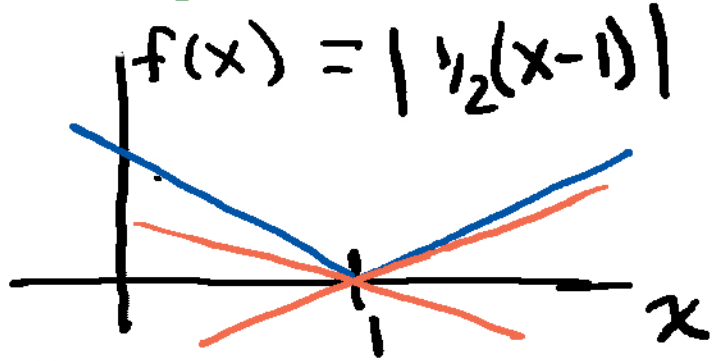
$$f(x) \geq f(x_0) + d(x_0)(x - x_0)$$

"above tangent line"

Subderivative (convex)

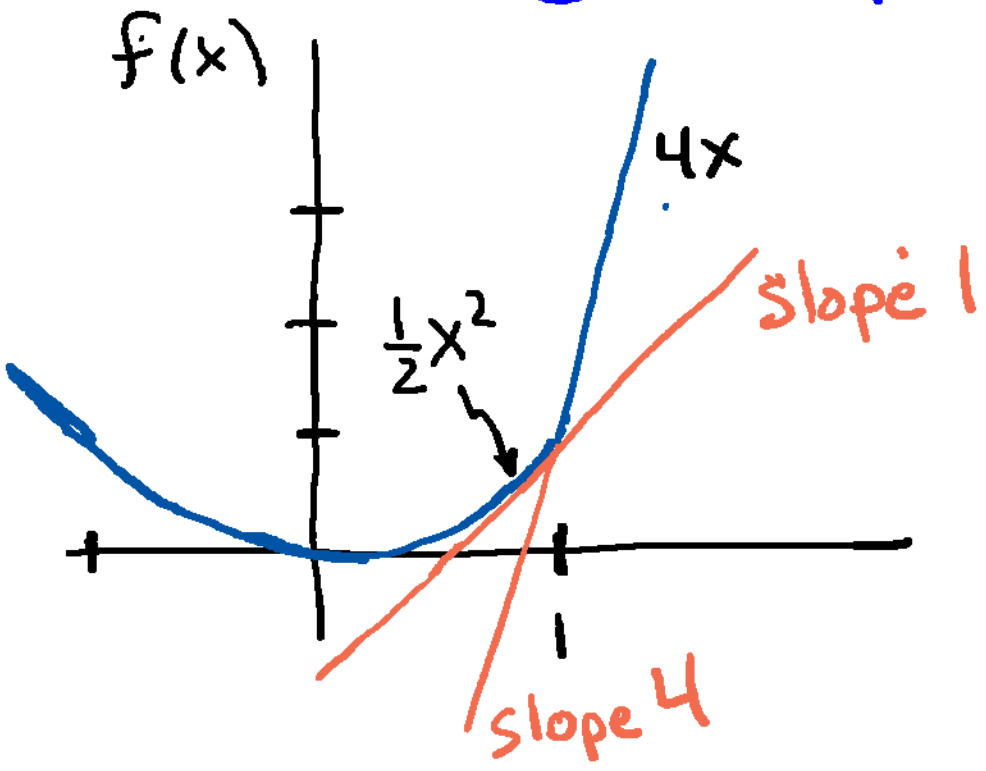
Any $d_s(x_0)$: $f(x) \geq f(x_0) + d_s(x_0)(x - x_0)$

$x < 1$: $d_s(x) = -\frac{1}{2}$; $x > 1$: $d_s(x) = \frac{1}{2}$



$-\frac{1}{2} \leq d_s(1) \leq \frac{1}{2}$ "below $f(x)$"

Subderivatives produce "reasonable" downhill directions



Example: $f(x) = \begin{cases} \frac{1}{2}x^2 & x<1 \\ 4x & x>1 \end{cases}$

convex

Subderivative

$d_s(x) = \begin{cases} x, & x<1 \\ 4 & x>1 \\ [1,4] & x=1 \end{cases}$

Subgradients generalize gradients
– Convex, nondifferentiable $\ell(\underline{w})$

Gradients –

$$\ell(\underline{w}) \geq \ell(\underline{w}_0) + (\underline{w}-\underline{w}_0)^T \underline{v}(\underline{w}_0) \qquad \underline{v}(\underline{w}) = \nabla_{\underline{w}} \ell(\underline{w})$$

"above tangent plane" $\left(\sum_{i=1}^{m} (w_i - w_{0i}) \frac{d}{dw_i} \ell(\underline{w}) \right)$

Subgradients –

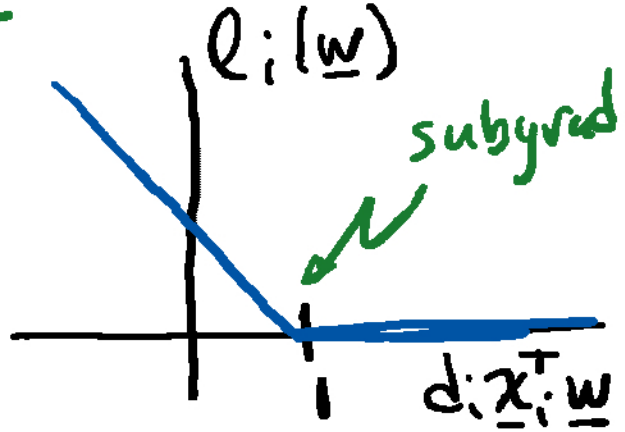
Any $\underline{v}(\underline{w})$: $\ell(\underline{w}) \geq \ell(\underline{w}_0) + (\underline{w}-\underline{w}_0)^T \underline{v}(\underline{w}_0)$

Gradient descent optimization: replace gradient with subgradient

Gradient descent for SVMs

$$\ell(\underline{w}) = \sum_{i=1}^{N} (1 - d_i \underline{x}_i^T \underline{w})_+ \rightarrow \text{subgradient}$$

$$\ell_i(\underline{w}) = (1 - d_i \underline{x}_i^T \underline{w})_+ = \begin{cases} 1 - d_i \underline{x}_i^T \underline{w} & d_i \underline{x}_i^T \underline{w} < 1 \\ 0 & d_i \underline{x}_i^T \underline{w} > 1 \end{cases}$$



Subgradient

$$\underline{v}_i(\underline{w}) = \begin{cases} -d_i \underline{x}_i & d_i \underline{x}_i^T \underline{w} < 1 \\ 0 & d_i \underline{x}_i^T \underline{w} \geq 1 \end{cases} = -d_i \underline{x}_i \mathbb{I}_{\{d_i \underline{x}_i^T \underline{w} < 1\}}$$

indicator function

Cost $f(\underline{w}) = \ell(\underline{w}) + \lambda \|\underline{w}\|_2^2$

$$\Rightarrow \nabla f(\underline{w})\big|_{\underline{w}^{(k)}} = \sum_{i=1}^{N} \left(-d_i \underline{x}_i \mathbb{I}_{\{d_i \underline{x}_i^T \underline{w}^{(k)} < 1\}}\right) + 2\lambda \underline{w}^{(k)}$$

Gradient descent $\underline{w}^{(k+1)} = \underline{w}^{(k)} - \tau \nabla f(\underline{w})\big|_{\underline{w}^{(k)}}$

Stochastic Gradient Descent

Objectives

- Simplify gradient descent update
- Common methods for cycling through data
- Benefits
- Examples

Stochastic gradient descent updates weights using part of the data

$$f(\underline{w}) = \ell(\underline{w}) + \lambda r(\underline{w}) \qquad \underline{w}^{(k+1)} = \underline{w}^{(k)} - \frac{\tau}{2}\nabla_{\underline{w}} f(\underline{w})$$

"loss" "regularize" gradient

squared error

$$\ell(\underline{w}) = \sum_{i=1}^{N} (d_i - \underline{x}_i^T \underline{w})^2$$

$$\nabla_{\underline{w}} \ell(\underline{w}) = -2\sum_{i=1}^{N} (d_i - \underline{x}_i^T \underline{w})\,\underline{x}_i$$

hinge loss

$$\ell(\underline{w}) = \sum_{i=1}^{N} (1 - d_i \underline{x}_i^T \underline{w})_+$$

$$\nabla_{\underline{w}} \ell(\underline{w}) = -\sum_{i=1}^{N} I_{\{d_i \underline{x}_i^T \underline{w} < 1\}}\,\underline{x}_i$$

$(d_i, \underline{x}_i), i = 1, \ldots, N$

labels, features

depends on all the data

SGD: $f(\underline{w}) = \sum_{i=1}^{N} f_i(\underline{w})$ Define i_k, $k = 1, 2, \ldots$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \frac{\tau}{2}\nabla_{\underline{w}} f_{i_k}(\underline{w}^{(k)})$$

depends on one sample $(d_{i_k}, \underline{x}_{i_k})$

SGD cycles through training data

1) Cyclical (incremental gradient descent)

$i_k = k \bmod N$ e.g. $i_k = 1,2,3,4,1,2,3,4,1,2,3\ldots$

2) Random permutation (reshuffle every N rounds)

$i_k = 2,4,1,3,\boxed{2,1,4,3},4,3,1,2\ \ldots$

3) Stochastic gradient descent (uniformly at random)

$i_k = \text{uniform}\{1,2,\ldots,N\}$ $i_k = 2,1,3,1,4,4,2,3,1,3\ \ldots$

Update by $-\tau_2 \nabla_{\underline{w}} f_{i_k}(\underline{w})$ at each iteration

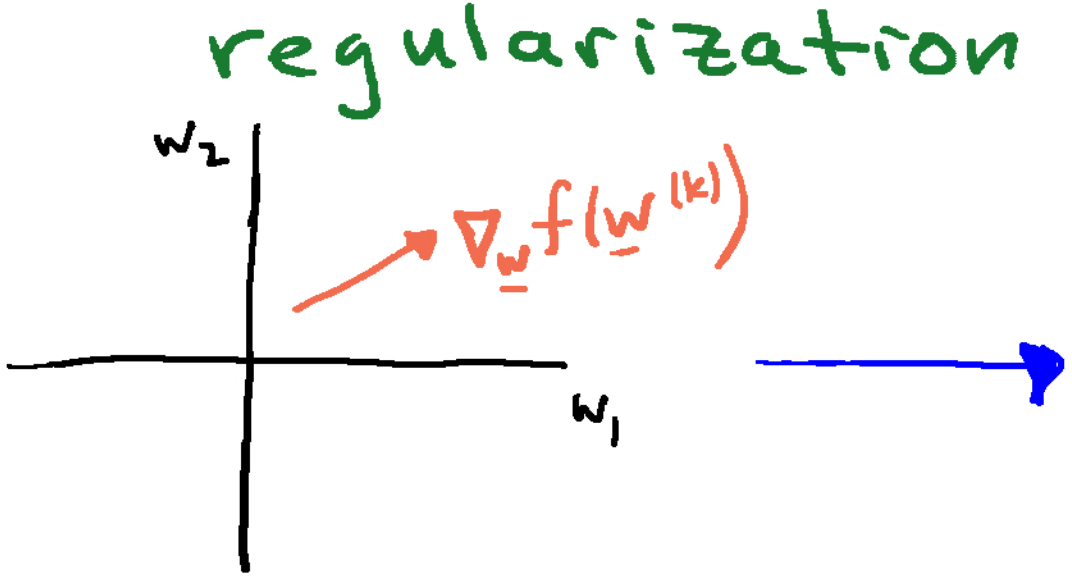
On average gives gradient $E\{\nabla_{\underline{w}} f_{i_k}(\underline{w})\} \simeq \frac{\nabla_{\underline{w}} f(\underline{w})}{N}$

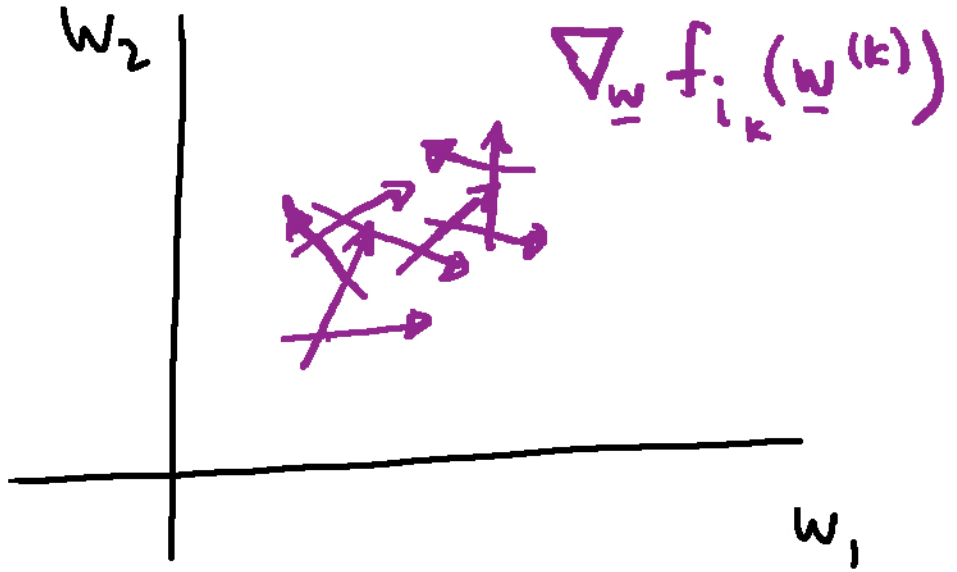
SGD has computational benefits

1) Computing $\nabla_{\underline{w}} f_{i_k}(\underline{w}^{(k)})$ is easier/faster than $\nabla_{\underline{w}} f(\underline{w}^{(k)})$

2) May not be able to store $\underline{x}_i$, $i=1,\dots,N$ in memory

3) Noisy gradient $\nabla_{\underline{w}} f_{i_k}(\underline{w}^{(k)})$ introduces added regularization





Example: Ridge Regression

$$f(\underline{w}) = \sum_{i=1}^{N} (d_i - \underline{x}_i^T \underline{w})^2 + \lambda \|\underline{w}\|_2^2 = \sum_{i=1}^{N} \underbrace{\left\{ (d_i - \underline{x}_i^T \underline{w})^2 + \frac{\lambda}{N} \|\underline{w}\|_2^2 \right\}}_{f_i(\underline{w})}$$

$$\nabla_{\underline{w}} f_i(\underline{w}) = \nabla_{\underline{w}} \left[(d_i - \underline{x}_i^T \underline{w})^2 + \frac{\lambda}{N} \underline{w}^T \underline{w} \right]$$

$$= -2 (d_i - \underline{x}_i^T \underline{w}) \underline{x}_i + \frac{2\lambda}{N} \underline{w}$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \frac{\tau}{2} \nabla_{\underline{w}^{(k)}} f_{i_k}(\underline{w}^{(k)})$$

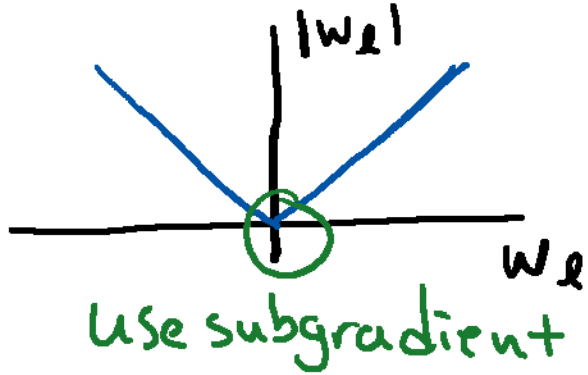
$$= \underline{w}^{(k)} + \tau (d_{i_k} - \underline{x}_{i_k}^T \underline{w}^{(k)}) \underline{x}_{i_k} - \frac{\tau\lambda}{N} \underline{w}^{(k)}$$

VS. $\underline{w}^{(k+1)} = \underline{w}^{(k)} + \tau \underline{A}^T (\underline{A} \underline{w}^{(k)} - \underline{d}) - \lambda\tau \underline{w}^{(k)}$ $\quad \underline{A}: N \times M$

Example: Gradient descent for LASSO

$$f(\underline{w}) = \sum_{i=1}^{N} (d_i - \underline{x}_i^T \underline{w})^2 + \lambda \|\underline{w}\|_1 = \sum_{i=1}^{N} \underbrace{\left\{ (d_i - \underline{x}_i^T \underline{w})^2 + \frac{\lambda}{N} \|\underline{w}\|_1 \right\}}_{f_i(\underline{w})}$$

Consider $\nabla_{\underline{w}} \sum_{\ell=1}^{M} |w_\ell|$



Use subgradient

$$\frac{d}{dw_\ell} |w_\ell| = \begin{cases} \text{sign}(w_\ell) & w_\ell \neq 0 \\ [-1, 1] & w_\ell = 0 \end{cases}$$

"0" popular

Write $\nabla_{\underline{w}} \|\underline{w}\|_1 = \text{sign}(\underline{w})$

$$\nabla_w f_i(\underline{w}) = -2(d_i - \underline{x}_i^T \underline{w}) \underline{x}_i + \frac{\lambda}{N} \text{sign}(\underline{w})$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} + \tau (d_{i_k} - \underline{x}_{i_k}^T \underline{w}^{(k)}) \underline{x}_{i_k} - \frac{\lambda \tau}{2N} \text{sign}(\underline{w}^{(k)})$$