CSC 580 Principles of Machine Learning

07 Linear models for classification

Chicheng Zhang

Department of Computer Science

THE UNIVERSITY OF ARIZONA®

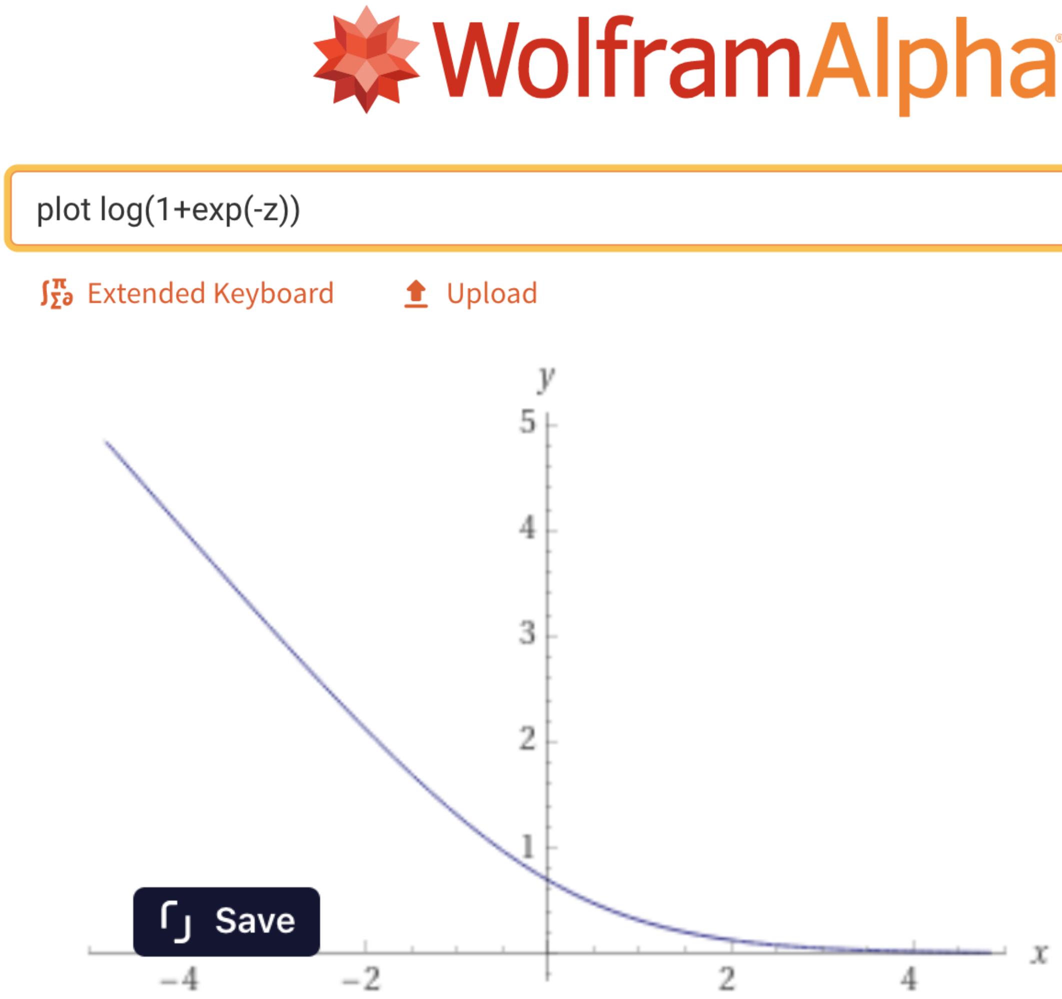
*slides credit: built upon CSC 580 Fall 2021 lecture slides by Kwang-Sung Jun

Classification with linear models

- Logistic loss
  - $x_i \in \mathbb{R}^d, \quad y_i \in \{1, -1\}$
  - $S = \{(x_i, y_i)\}_{i=1}^n$
  - $\ell(w; x_i, y_i) = \log(1 + \exp(-y_i \cdot w^\top x_i))$

- The ERM principle, again!
  $\hat{w} = \text{argmin}_{w \in \mathbb{R}^d} F(w), \ F(w) := \sum_{i=1}^n \ell(w; x_i, y_i)$

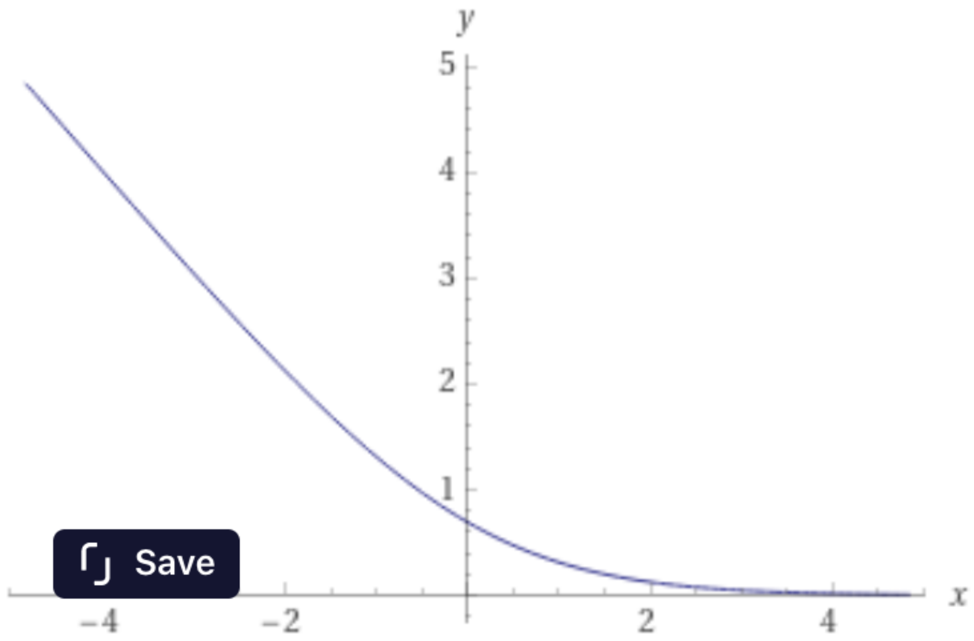
- How to optimize?



$y_i w^\top x_i$

First, is it convex?

- How do we check the convexity of F?
  - Is $\ell(w; x_i, y_i) = \log(1 + \exp(-y_i \cdot w^\top x_i))$ convex in w?
  - Observation: $\ell(w; x_i, y_i) = h(y_i \cdot w^\top x_i)$ where $h(z) = \log(1 + \exp(-z))$
  - It suffices to check that $h(z)$ is convex
  - Indeed, $h''(z) = \frac{e^{-z}}{(1+e^{-z})^2} \geq 0$

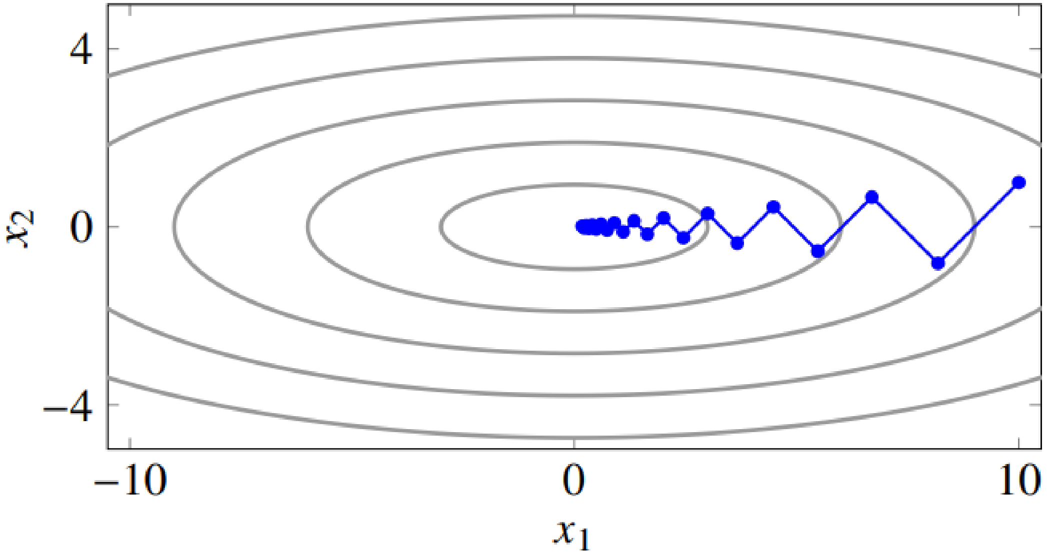


- Alternative route: check the PSD-ness of $\nabla^2 \ell(w; x_i, y_i)$

- Great! Let's solve $\nabla F(w) = 0$

Finding the minimizer of F: gradient descent

- Algorithm
  Input: initial point $w_0 \in \mathbb{R}^d$
  step sizes $\{\eta_t\}_{t=1}^{\infty}$
  stopping tolerence $\epsilon > 0$
  For $t = 1, \dots,$ max_iter
  - $w_t \leftarrow w_{t-1} - \eta_t \cdot \nabla F(w_{t-1})$
  - stop if $\left|\frac{F(w_t) - F(w_{t-1})}{F(w_{t-1})}\right| \le \epsilon$



Hyperparameters

- w_0: set it to 0
  - warmstart possible if you have a good guess
- stepsize
  - constant scheme: $\eta_t = \eta, \forall t$
  - $\eta_t = \frac{1}{\sqrt{t}}$
  - $\eta_t = \frac{1}{t}$
  - Line search possible
- ϵ: 10^{-4} to 10^{-7}… more of engineering.

http://www.seas.ucla.edu/~vandenbe/236C/lectures/gradient.pdf

More iterative methods

Algorithms	Number if iterations until convergence	Time complexity per iteration
Newton's method	Very small	nd^3
LBFGS	small	nmd
Gradient descent (GD)	large	nd
Stochastic gradient descent (SGD)	Very large	d

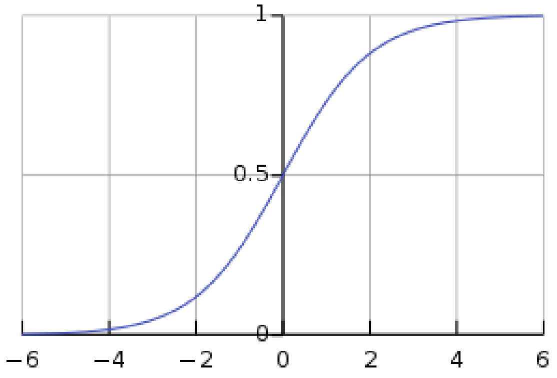
- n: #training examples
- d: dimensionality
- m: LBFGS's memory hyperparameter
- Will come back to SGD in later part of this lecture

Probabilistic interpretation of logistic regression

- How did they come up with the logistic loss?
- Let us begin using 1/0 encoding for the label (then later turn into 1/-1 encoding)

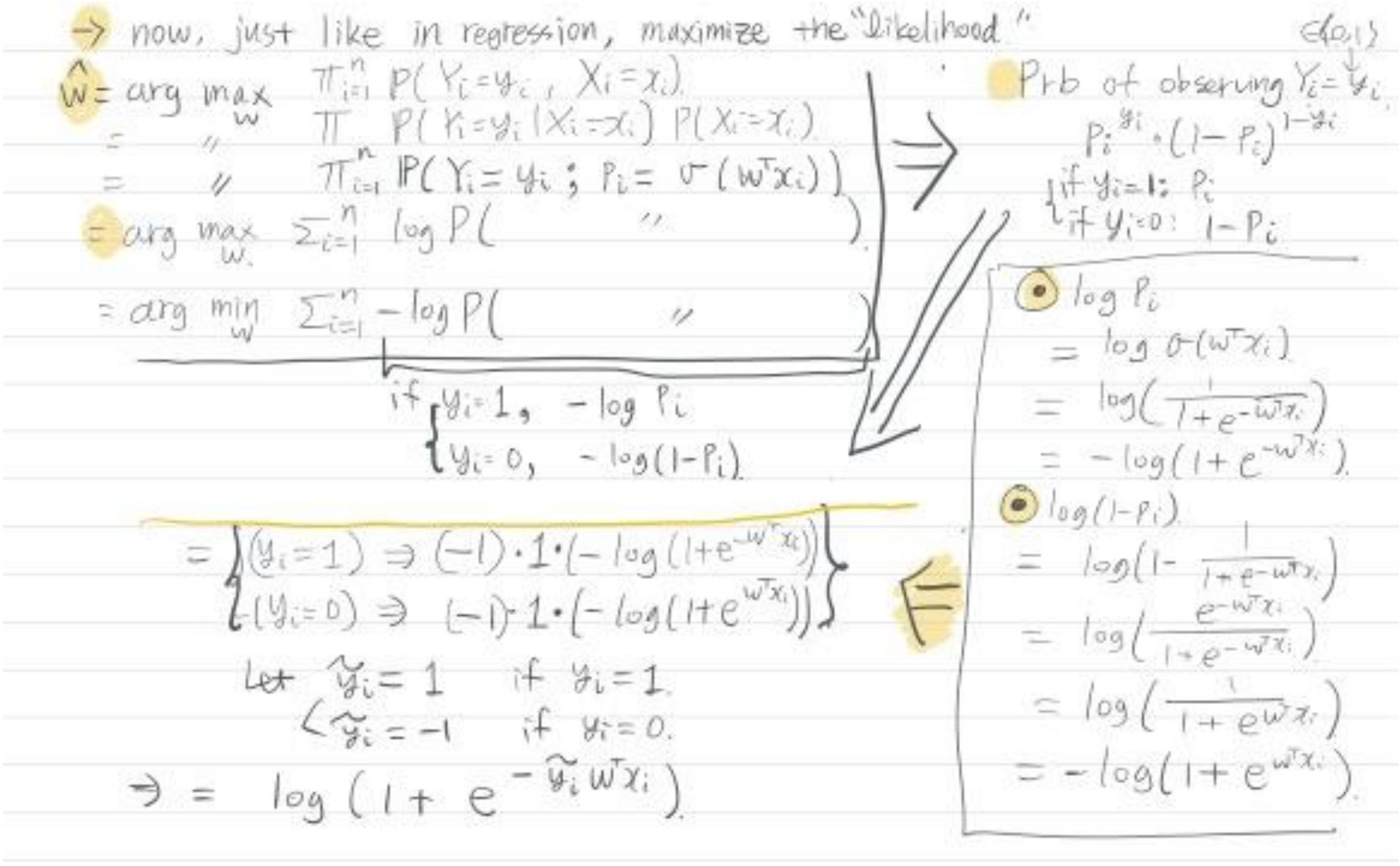
- $y_i \mid x_i \sim \mathrm{Bernoulli}(p_i)$, where $p_i = g(x_i)$

- Modeling attempt 1: $g(x_i) = w^\top x_i$
- Modeling attempt 2: $g(x_i) = \sigma(w^\top x_i)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function
  - i.e. logit $\log\left(\frac{p_i}{1-p_i}\right) = w^\top x_i$



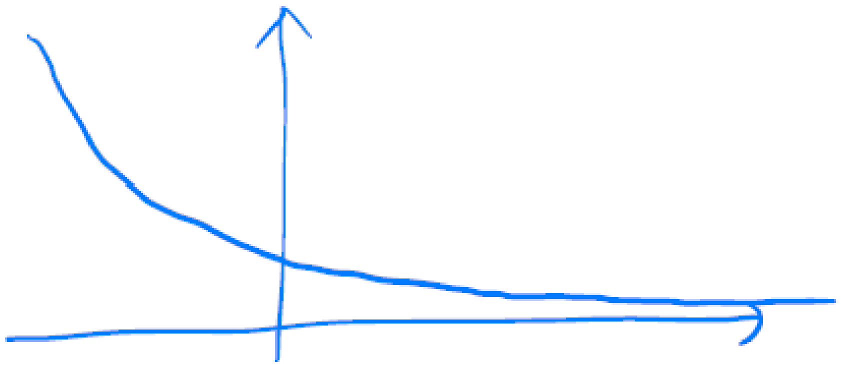
Probabilistic interpretation of logistic regression

Logistic regression as maximum likelihood estimation $\quad y_i \mid x_i \sim \mathrm{Bernoulli}(\sigma(w^\top x_i))$

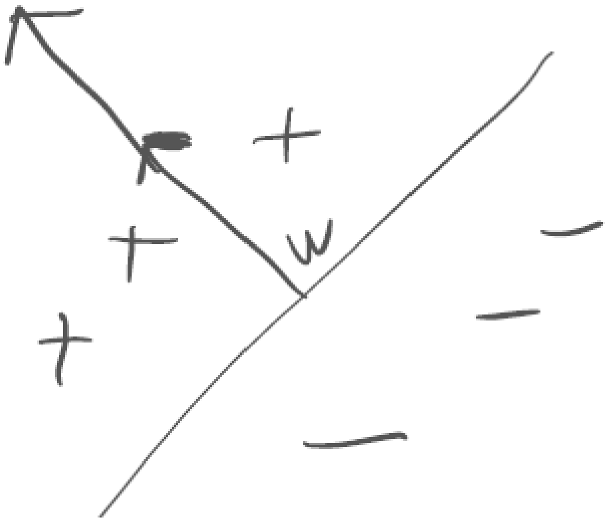


Caveat: Logistic regression may not have a minimizer without a regularizer

- E.g.,
  - training set has only one data point



  - more generally, linearly separable data.



  - Structure of minimizers, optimization properties discussed in

Convex Analysis at Infinity: An Introduction to Astral Space

Miroslav Dudík, Ziwei Ji, Robert E. Schapire, Matus Telgarsky

  - Adding regularization addresses this issue:
    $\hat{w} = \text{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(w; x_i, y_i) + \lambda \|w\|_2^2$

Next class (9/26)

- Dual of SVM; induced practical optimization algorithms
- Kernel methods
- Plan to release HW2
- Assigned reading: CIML 11.1-11.2

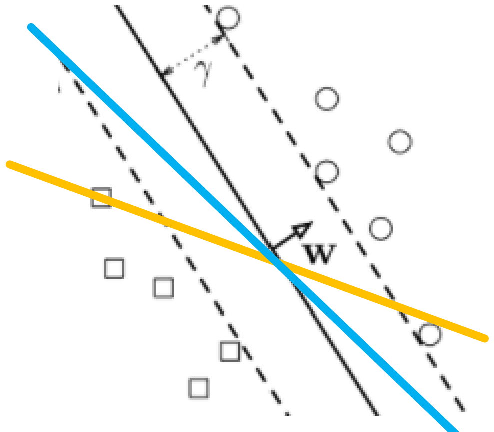
Support Vector Machines

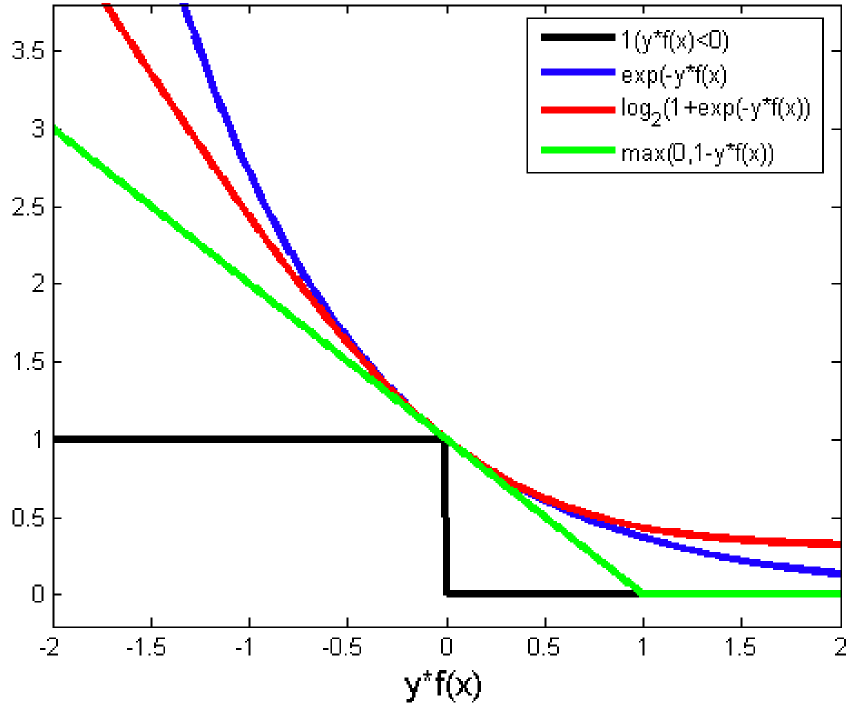
- In a nutshell
  - Perform regularized ERM $\hat{w} = \text{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(w; x_i, y_i) + \lambda \|w\|_2^2$ with the loss

$$\ell(w; x, y) = (1 - y \cdot w^\top x)_+$$

hinge loss

    - notation: $(z)_+ \coloneqq \max\{0, z\}$
- Interesting aspects
  - Works well in general
  - No corresponding probabilistic motivation
  - Geometric Interpretation: **maximize the margin**.





https://rohanvarma.me/Loss-Functions/

Remaining parts of the lecture

- Q1: How is the loss function motivated and how is it maximizing the margin?

- Q2: How to solve the SVM optimization problem efficiently?

SVM: motivation

- The goal of linear classifier: Find w so that the rule $h_w(x) = \mathrm{sign}(w^\top x)$ will have small generalization error $\mathrm{err}(h_w)$.

- ERM: it seems natural to use the loss $1\{h_w(x) \neq y\}$, but…
  - NP-hard (e.g. Guruswami and Raghavendra, 2009)
  - There might be multiple minima. How to break ties?

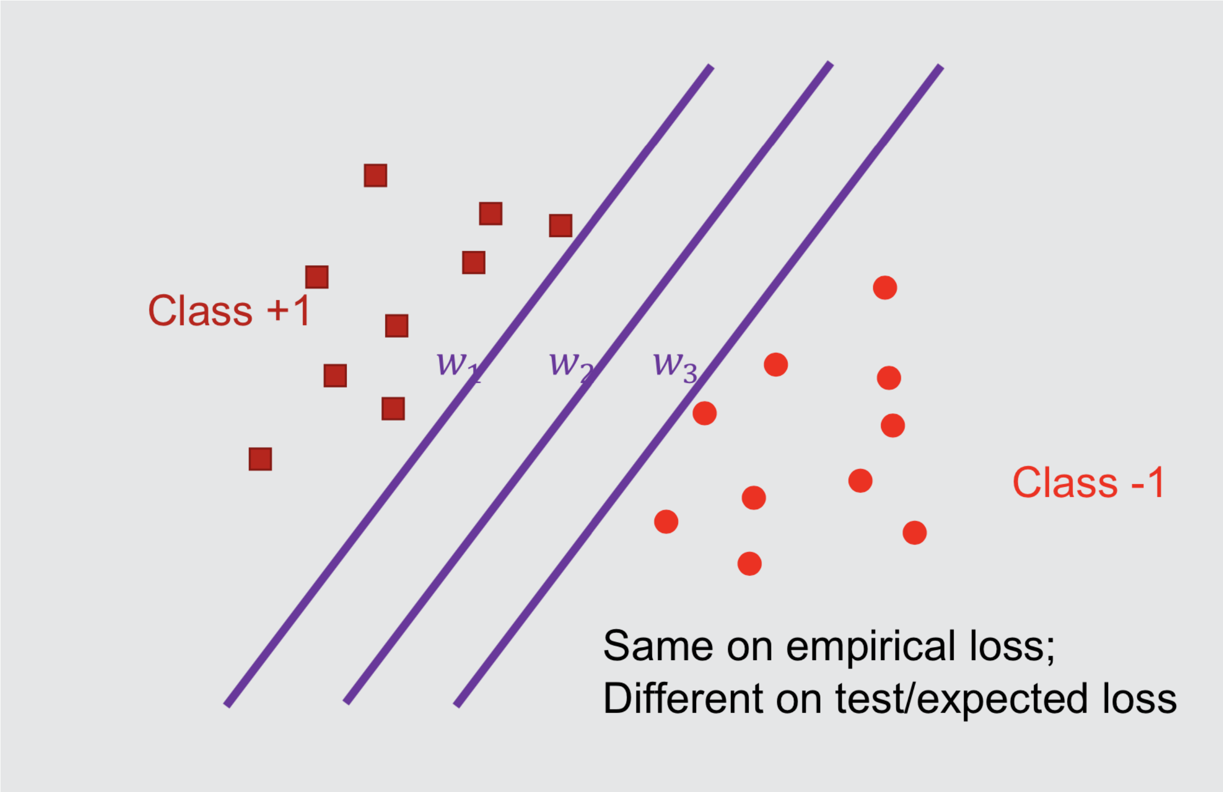
- Okay, we're stuck. Let us consider a **simple problem** and then try to extend it to the generic problem.
- The simple case: **linearly separable data** (recall perceptron)

https://www.cs.cornell.edu/courses/cs6781/2020sp/lectures/10_hardness2.pdf

Linearly separable data

- Recall: we can minimize 0-1 loss here with a reasonable time complexity!
    - e.g., run perceptron until it classifies train set perfectly
- But, among these minimizers, which one should we pick?

- Idea: pick the hyperplane such that its distances to all training examples are far

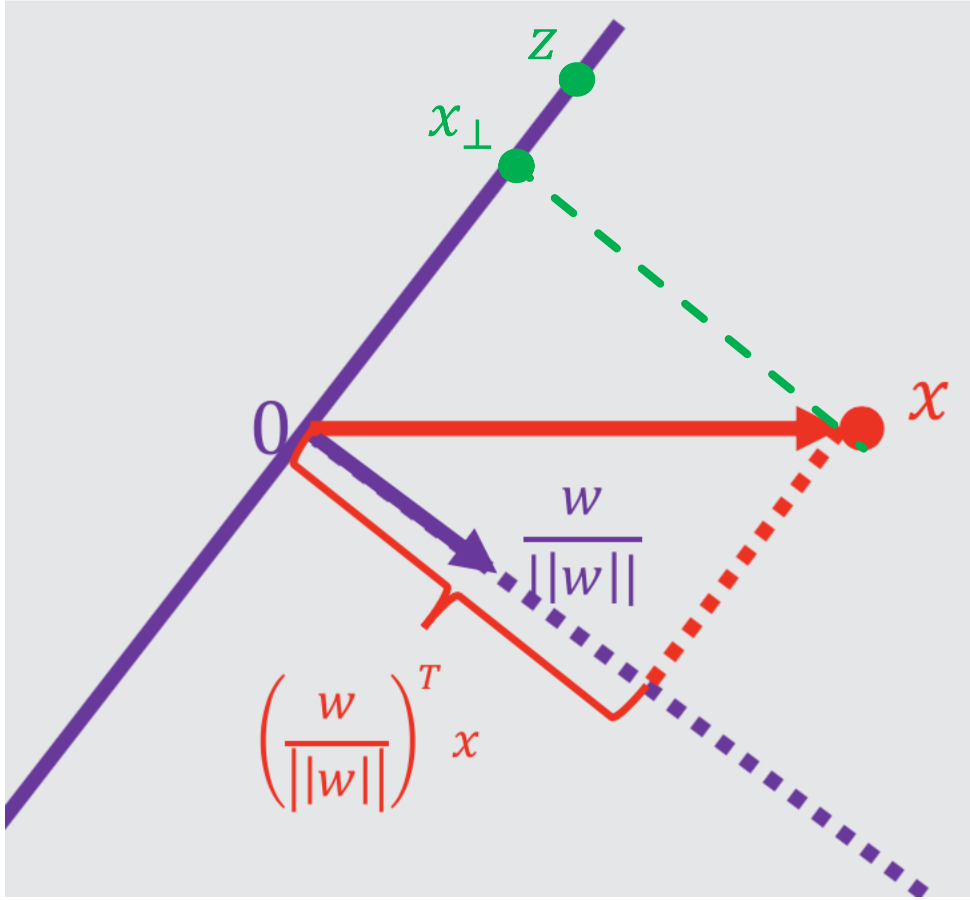


Facts on vectors

- (Lem 1) a vector x has distance $\frac{w^\top x}{\|w\|}$ to the hyperplane $w^\top x = 0$

- How about with bias? $w^\top x + b = 0$
- Let us be explicit on the bias: $f(x; w, b) = w^\top x + b$

- recall: w is orthogonal to the hyperplane $w^\top x + b = 0$
  - why? (left as exercise)

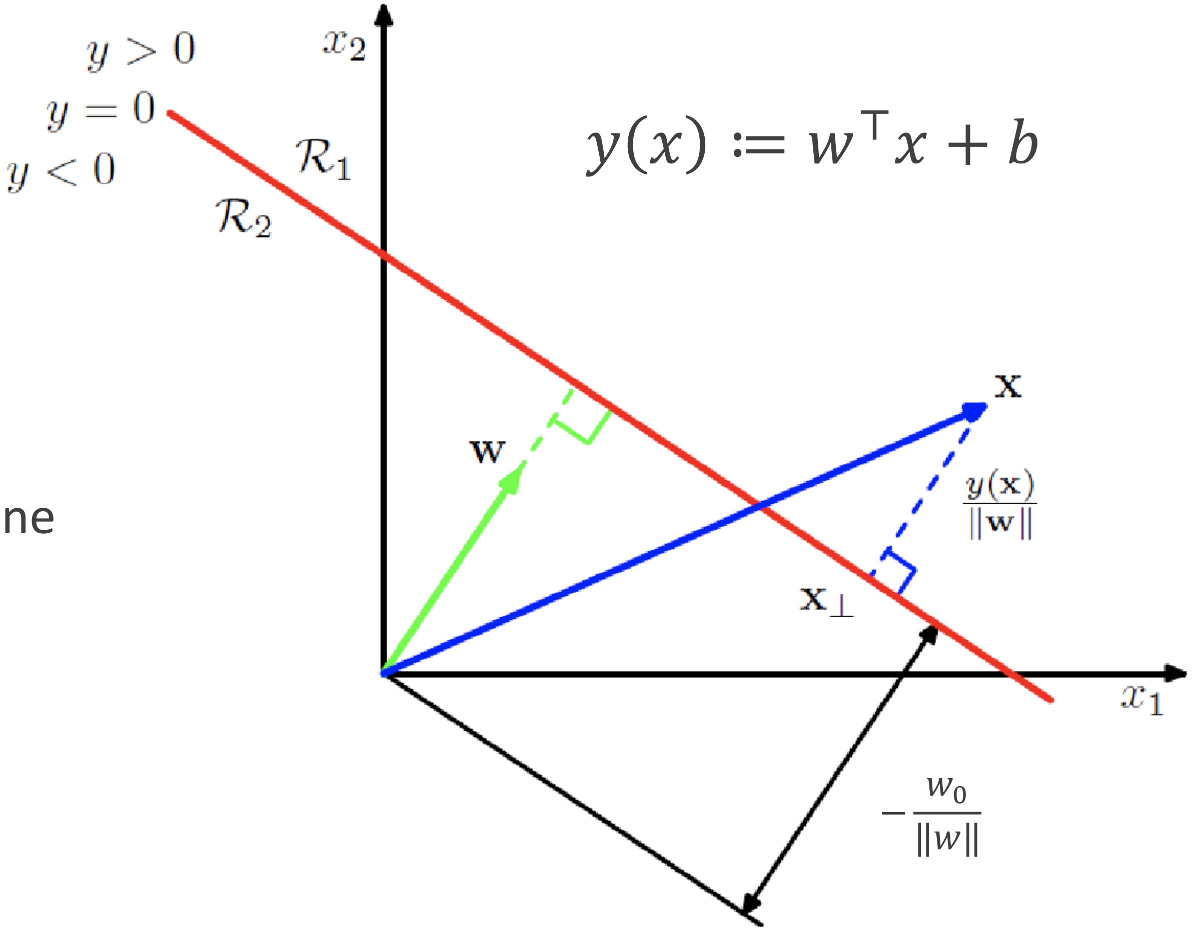


Facts on vectors

- (Lem 2) x has distance $\frac{|w^\top x+b|}{\|w\|}$ to the hyperplane $w^\top x + b = 0$

claim1 : x can be written as $x = x_\perp + r\frac{w}{\|w\|}$ where $x_\perp$ is the projection of x onto the hyperplane.

claim2 : then, $|r|$ is the distance between x and the hyperplane

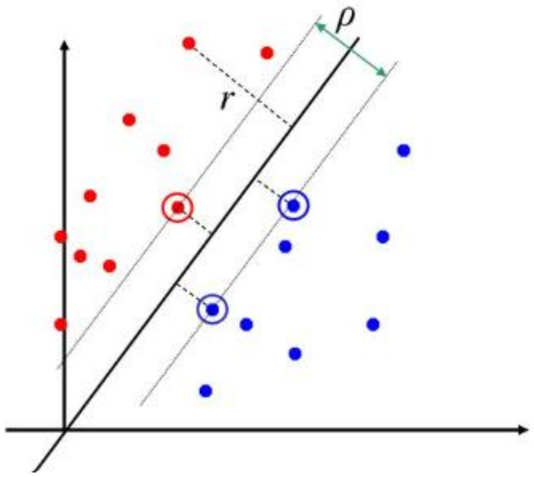


Solving for r: $w^\top x + b = w^\top x_\perp + r\frac{w^\top w}{\|w\|} + b = r\|w\|$.

this implies $|r| = \frac{|w^\top x+b|}{\|w\|}$

Figure from *Pattern Recognition and Machine Learning*, Bishop

SVM derivation (1)



- Margin of (w, b) over all training points: $\gamma'(w,b) = \min_i \frac{|w^\top x_i + b|}{\|w\|}$

- Choose (w, b) with the maximum margin? .. wait, we also want it to be a perfect classifier
  - redefine it

$$\gamma(w,b) = \min_i \frac{y_i(w^\top x_i + b)}{\|w\|}$$

- Choose w with the maximum margin (and perfect classification)

$$(\hat{w}, \hat{b}) = \max_{w,b} \min_{i=1}^{n} \frac{y_i(w^\top x_i + b)}{\|w\|}$$

  - One more issue: still, infinitely many solutions..!

SVM derivation (2)

$$(\hat{w}, \hat{b}) = \max_{w,b} \min_{i=1}^{n} \frac{y_i(w^\top x_i + b)}{||w||}$$

- Infinitely many solutions..
- It's actually a matter of removing 'duplicates'; ∃ many (w,b)'s that actually represent the same hyperplane.
- Quick solution = achieves the smallest margin
  - For any solution $(\hat{w}, \hat{b})$, let x_{i^*} be the **closest to the hyperplane** $\hat{w} x_i + \hat{b} = 0$
  - Imagine rescaling $(\hat{w}, \hat{b})$ so that $|\hat{w}^\top x_{i^*} + \hat{b}| = 1$
- We can always do that, but can we find a formulation that automatically finds that modified solution?
  - add the constraint $\min_i y_i(w^\top x_i + b) = 1$

SVM derivation (3)

$$\max_{w,b} \min_{i=1}^{n} \frac{y_i(w^\top x_i + b)}{\|w\|}$$
$$s.t.\ \min_{\mathrm{i}} y_i(w^\top x_i + b) = 1$$

- Summary: the constraint encodes (1) correct classification (2) there are no two solutions that represent the same hyperplane!
  - Note: If $(\hat{w}, \hat{b})$ is a solution, then the margin is $\frac{1}{\|\hat{w}\|}$

$$\max_{w,b} \frac{1}{\|w\|} \quad s.t.\ \min_{\mathrm{i}} y_i(w^\top x_i + b) = 1$$

$$\max_{w,b} \frac{1}{\|w\|} \quad s.t.\ \min_{i} y_i(w^\top x_i + b) \geq 1$$

(turns out to be equivalent..)

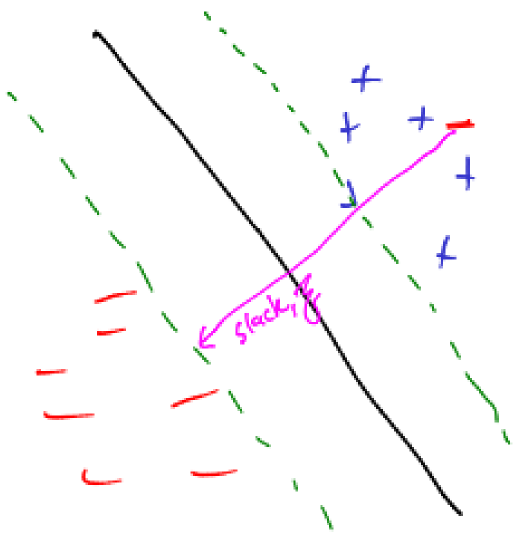
$$\max_{w,b} \frac{1}{\|w\|} \quad s.t.\ y_i(w^\top x_i + b) \geq 1, \forall i$$

Final formulation in the linearly separable setting: (quadratic programming)

$$\min_{w,b} \|w\|^2$$
$$\text{s.t.}\ y_i(w^\top x_i + b) \geq 1, \forall i$$

SVM in the nonseparable setting: Soft-margin

$$\min_{w,b} \|w\|^2$$
$$s.t.\ \ y_i(w^\top x_i + b) \geq 1, \forall i$$



- What if data is linearly nonseparable?
- Introduce 'slack' variables

$$\min_{w,b,\{\xi_i \geq 0\}} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \qquad \text{// } C \text{ is a hyper-parameter}$$
$$s.t.\ \ y_i(w^\top x_i + b) \geq 1 - \xi_i, \forall i$$

- Again, a quadratic programming problem
- Fix any w, b, the optimal ξ?

$\xi_i = 0$ if $y_i(w^\top x_i + b) \geq 1$, and $\xi_i = 1 - y_i(w^\top x_i + b)$

$$\min_{w,b} \ \|w\|^2 + C \sum_{i=1}^{n} \left(1 - y_i(w^\top x_i + b)\right)_+ \Leftrightarrow \text{Regularized hinge loss minimization } \lambda = \frac{1}{C}$$

Solving SVM optimization problems

- Two popular methods

- Method 1: stochastic gradient descent

- Method 2: solve the *dual problem* and transform the dual solution back

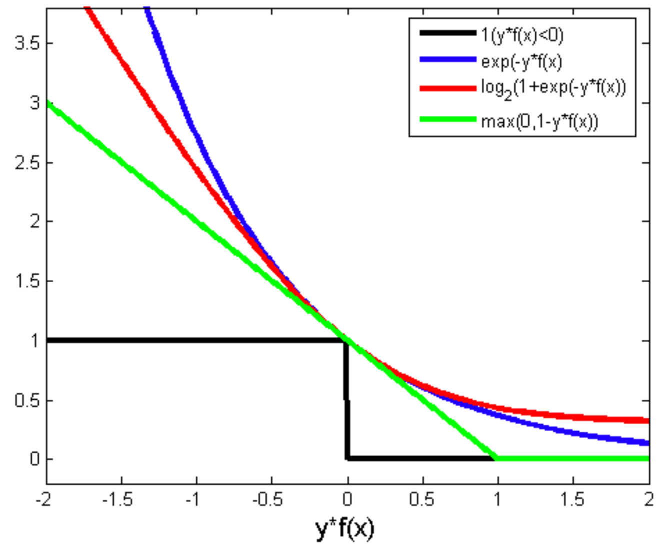
Stochastic gradient descent (SGD)

- Finding $\widehat{w} = \mathrm{argmin}_{w \in \mathbb{R}^d} F(w),\ F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$,

  where $f_i(w)$ is convex + quadratic, e.g.

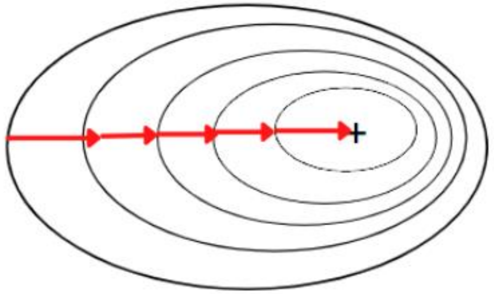
  $(1 - y_i\langle w, x_i\rangle)_+ + \lambda\|w\|_2^2$,

  $\log(1 + \exp(-y_i \cdot w^\top x_i)) + \lambda\|w\|_2^2$



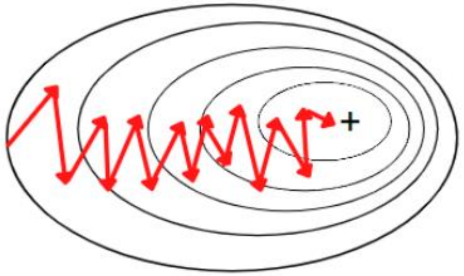
- Observation: gradient descent is computationally expensive
  - calculating exact gradient $\nabla F(w)$ takes at least $\Omega(n)$ time

Batch Gradient Descent



- Key idea (Robbins-Monro'51): descend in directions that are in-expectation $\nabla F(w)$
- For $t = 1, 2, \ldots, T$:
  - Choose $i_t \sim \mathrm{Uniform}(\{1, \ldots, n\})$
  - $w_{t+1} \leftarrow w_t - \eta_t \nabla f_{i_t}(w)$
- Output: (1) $\overline{w}_T := \frac{1}{T}\sum_{t=1}^{T} w_t$ (average iterate); (2) w_T (last iterate)

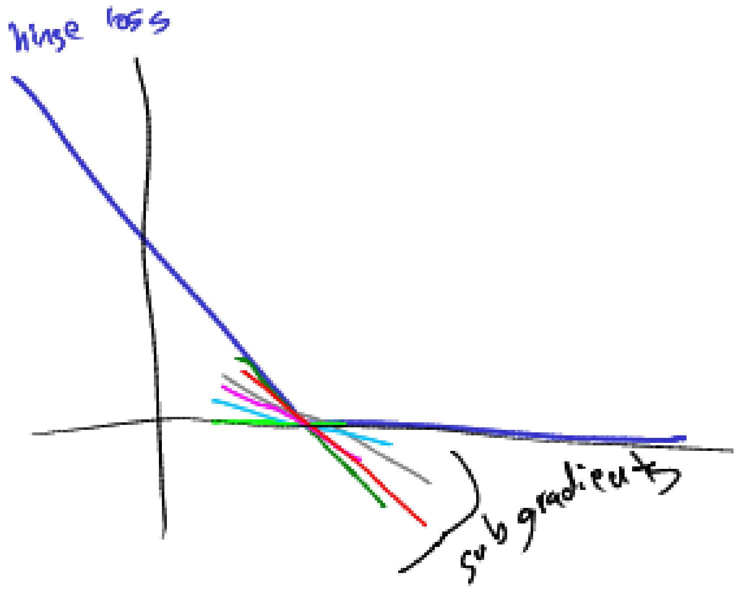
Stochastic Gradient Descent



SGD: handling nondifferentiable objectives

- Hinge loss:

  $f(w) = h(w) + \frac{\lambda}{2}\|w\|_2^2$, where $h(w) = (1 - y\langle w, x\rangle)_+$



- For some w, $\nabla h(w)$ does not exist (say, d=1)

- Workaround: descent in the *subgradient* direction

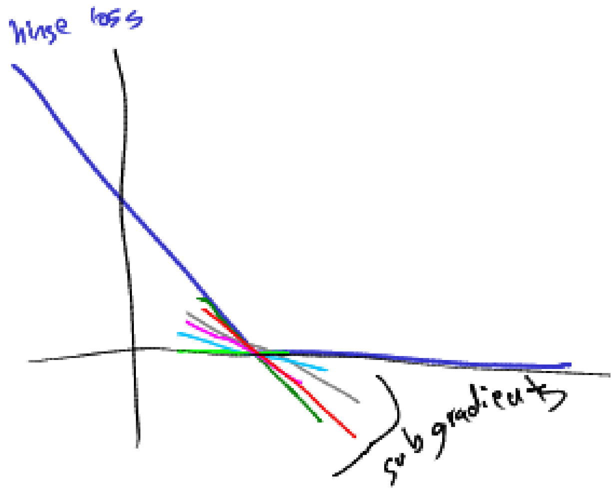
- [Def] For convex function h, $g \in \mathbb{R}^d$ is said to be a subgradient of h at w, if for any u,

$$h(u) \geq h(w) + \langle g, u - w\rangle$$

  The set of subgradients of h at w is denoted as $\partial h(w)$

- For differentiable h, $\partial h(w) = \{\nabla h(w)\}$

Subgradient: intuition and properties



- Example: $h(w) = (1-w)_+$,

$$\partial h(w) = \begin{cases} \{-1\}, & w < 1 \\ [-1,0], & w = 1 \\ \{0\}, & w > 1 \end{cases}$$

- (Lem) If $h(w) = l(\langle a, w\rangle + b)$ for some convex l on $\mathbb{R}$, and suppose $z \in \partial l(\langle a, w\rangle + b)$. Then, $az \in \partial h(w)$
  - Generalizes chain rule of differentiation

- Practical implication: For $f(w) = (1 - y\langle w, x\rangle)_+$, the following vector(s) are in $\partial f(w)$ (and are thus valid descent directions):

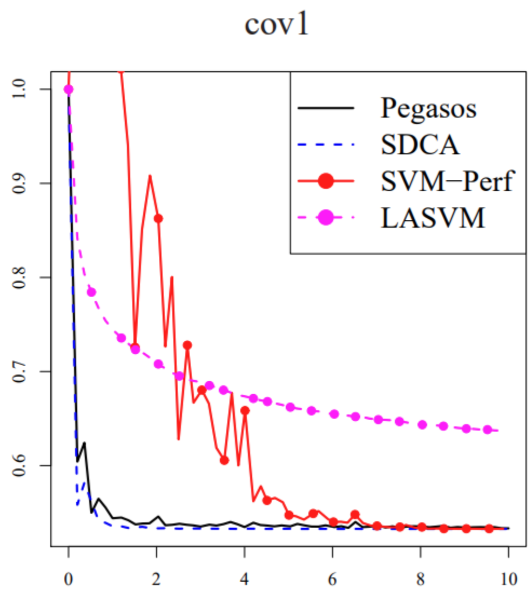
$$\begin{cases} -yx, & y\langle w, x\rangle < 1 \\ -uyx \text{ for } u \in [0,1], & y\langle w, x\rangle = 1 \\ 0, & y\langle w, x\rangle > 1 \end{cases}$$

SGD: convergence guarantee

- (Thm) Suppose $F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$, where $f_i(w) = h_i(w) + \lambda\|w\|_2^2$, and $h_i(w)$ is L-Lipschitz, then SGD with step size $\eta_t = \frac{1}{\lambda t}$ satisfies that

$$\mathbb{E}[F(\bar{w}_T)] - \min_w F(w) \leq O\left(\frac{L^2 \log T}{\lambda T}\right),$$

where $\bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$



- [Def] h is said to be L-Lipschitz, if for any u, v, $|h(u) - h(v)| \leq L\|u - v\|_2$

- $\tilde{O}\left(\frac{1}{T}\right)$ rate; if target optimization precision ϵ, then $O\left(\frac{1}{T}\right) \leq \epsilon \Leftarrow T \geq O\left(\frac{1}{\epsilon}\right)$
- Larger λ, "Smoother" $h_i \Longrightarrow$ easier to optimize

Shalev-Shwartz, Singer, Srebro, Cotter, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM", 2011

Solving SVM optimization problems

- Two popular methods

- Method 1: stochastic gradient descent

- Method 2: solve the *dual problem* and transform the dual solution back

Constrained optimization and Lagrange multiplier

- Lagrange multiplier: a powerful tool for solving *constrained* optimization problems.

$$\min_w f(w)$$
$$s.t. \quad g_i(w) \le 0, \forall i = 1, \dots, k$$
$$h_j(w) = 0, \forall j = 1, \dots, \ell$$

- Lagrangian: $\mathcal{L}(w, \alpha, \beta) \coloneqq f(w) + \sum_i \alpha_i g_i(w) + \sum_j \beta_j h_j(w)$, where α_i, β_j's are Lagrange multipliers
- Define $\theta_P(w) \coloneqq \max_{\alpha,\beta:\alpha_i \ge 0, \forall i} \mathcal{L}(w, \alpha, \beta)$
- (Thm) $\theta_P(w) = \begin{cases} f(w), & \text{if } w \text{ satisfies all the constraints} \\ +\infty, & \text{otherwise} \end{cases}$
- This implies that solving the following *unconstrained* problem is equivalent to solving the original constrained problem!

$$\min_w \theta_P(w) = \min_w \max_{\alpha,\beta:\alpha_i \ge 0, \forall i} \mathcal{L}(w, \alpha, \beta)$$

The dual problem

- Why dual?
  - Alternative way of efficient optimization
  - Gives rise to "kernel trick"

Recall: $p^* := \min_w \theta_P(w) = \min_w \max_{\alpha_1,\ldots,\alpha_k \geq 0, \beta_1,\ldots,\beta_\ell} \mathcal{L}(w, \alpha_{1:k}, \beta_{1:\ell})$

- Dual problem: $d^* := \max_{\alpha_1,\ldots,\alpha_k \geq 0, \beta_1,\ldots,\beta_\ell} \min_w \mathcal{L}(w, \alpha_{1:k}, \beta_{1:\ell})$
- [Def] "Strong duality holds": $p^* = d^*$
- To satisfy strong duality, we need conditions:
  - (1) f and g's are convex. h's are affine.
  - (2) Slater's condition: $\exists$ feasible point x_0: $g_i(x_0) < 0, i = 1, \ldots, k$

- For more properties, see e.g. Lieven Vandenberghe's lecture on convex optimization duality

Dual problem for homogeneous SVM

$$\min_w \frac{1}{2}\|w\|^2$$
$$s.t.\ \ y_i w^\top x_i \geq 1, \forall i$$

$$\mathcal{L}\,(w, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i w^\top x_i - 1)$$

- Claim: the dual problem is $\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$

- Proof idea: the dual problem is $\max_{\alpha \geq 0} \min_w \mathcal{L}(w, \alpha)$; fix any α, the optimal w is such that

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \implies w = \sum_i \alpha_i y_i x_i$$

Dual problem for nohomogeneous SVM

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ \ y_i(w^\top x_i + b) \geq 1, \forall i$$

$$\mathcal{L}\left((w,b),\alpha\right) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(w^\top x_i + b) - 1)$$

- Claim: the dual problem is $$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$
  s.t. $\sum_i \alpha_i y_i = 0$

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_i \alpha_i y_i = 0$$

Using the same reasoning as previous slide, you should be able to prove the claim!

The optimality condition

- From now on, suppose the strong duality holds.
- Then, $w^*, (\alpha^*, \beta^*)$ are optimal solutions to the primal and dual problems $\Leftrightarrow$
  $w^*, (\alpha^*, \beta^*)$ satisfy the following Karush-Kuhn-Tucker (KKT) condition

Feasibility

$$\alpha_i^* \geq 0, i = 1, \dots, k$$
$$g_i(w^*) \leq 0, i = 1, \dots, k$$
$$h_j(w^*) = 0, j = 1, \dots, \ell$$

Stationarity

$$\frac{\partial \mathcal{L}}{\partial w}(w^*, \alpha^*, \beta^*) = 0$$

Complementary slackness

$$\alpha_i^* g_i(w^*) = 0, i = 1, \dots, k$$

- Implication: this links the primal optimal w^* to the dual optimal (α^*, β^*)
  - Enables recovery of near optimal w from near-optimal (α, β)

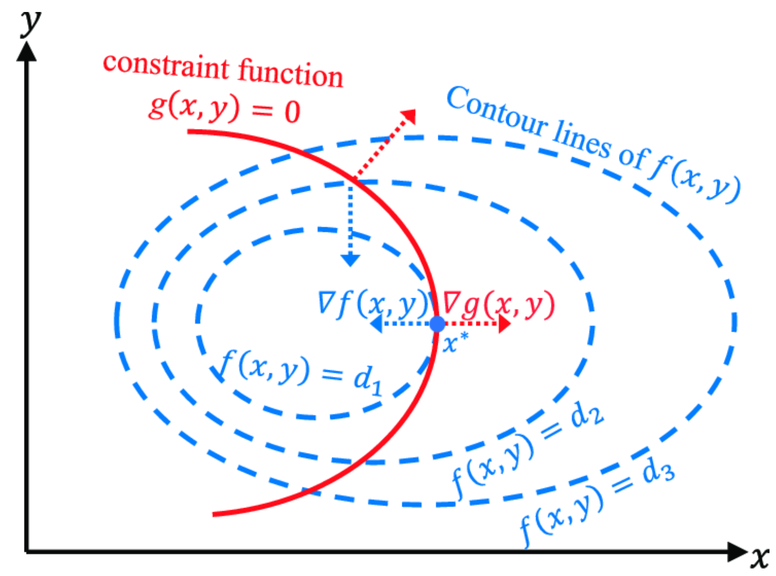
Optimality condition: stationarity

w^*, the solution of

$$\min_{w} f(w) \quad s.t. \quad h(w) = 0$$

satisfies that $\nabla\mathcal{L}(w^*, \beta^*) = 0$ for some β^*, i.e.

$$\nabla f(w^*) = -\beta^* \, \nabla h(w^*)$$



Key idea: if $\nabla f(w^*)$ is not colinear with $\nabla h(w^*)$ $\Rightarrow$ can locally decrease f while staying in $h(w) = 0$

Ex: $f(w) = w_1^2 + w_2^2, \; h(w) = w_1 + w_2 - 1$

Optimal solution w^* satisfies: $(2w_1^*, 2w_2^*) = -\beta^*(1,1) \Rightarrow w_1^* = w_2^*$

Optimality condition: complementary slackness

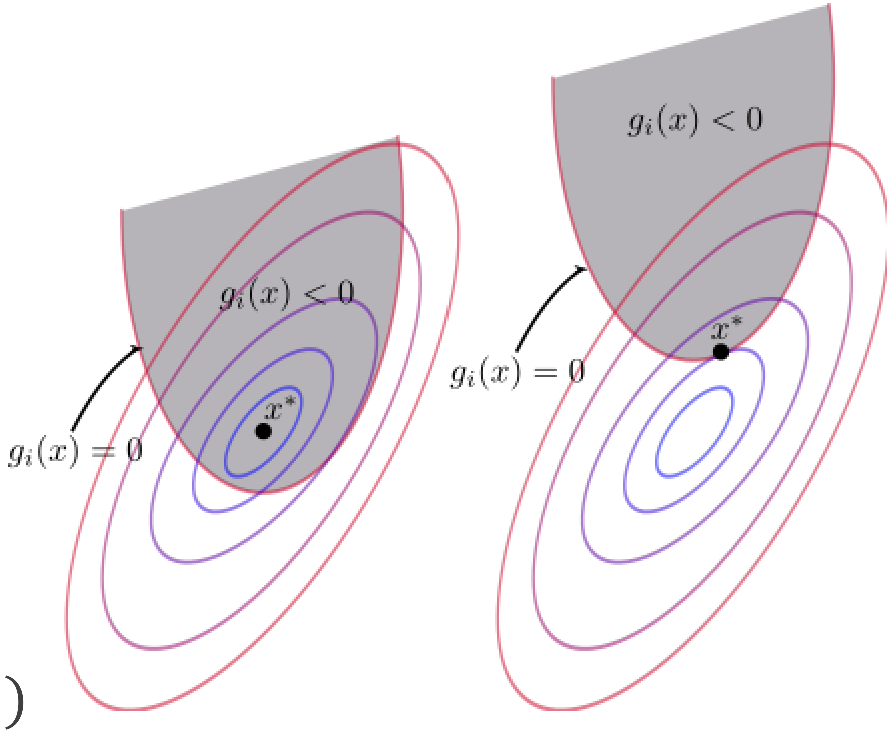
- w^*, the solution of

$$\min_w f(w)$$
$$s.t. \quad g(w) \le 0$$

satisfies that, there exists some dual variable $\alpha^* \ge 0$, s.t.

(1) $\nabla \mathcal{L}(w^*, \alpha^*) = 0$ for some, i.e. $\nabla f(w^*) = -\alpha^* \nabla g(w^*)$

(2) $\alpha^* \cdot g(w^*) = 0$



- Case 1: $g(w^*) < 0 \Rightarrow \alpha^* = 0 \Rightarrow \nabla f(w^*) = 0$
- Case 2: $g(w^*) = 0 \Rightarrow \nabla f(w^*)$ needs to be colinear with $\nabla g(w^*)$

The dual problem

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ \ y_i(w^\top x_i + b) \geq 1, \forall i$$

$$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$
s.t. $\sum_i \alpha_i y_i = 0$

- Quadratic programming
- Affine constraints
- n variables vs d+1 variables
- Why bother with n variables?

- How to get back the primal solution?
- Use optimality condition:

$$\frac{\partial \mathcal{L}}{\partial w}(w^*, \alpha^*) = w^* - \sum_{i=1}^{n} \alpha_i^* y_i x_i = 0$$
$$\Longrightarrow w^* = \sum_i \alpha_i^* y_i x_i$$

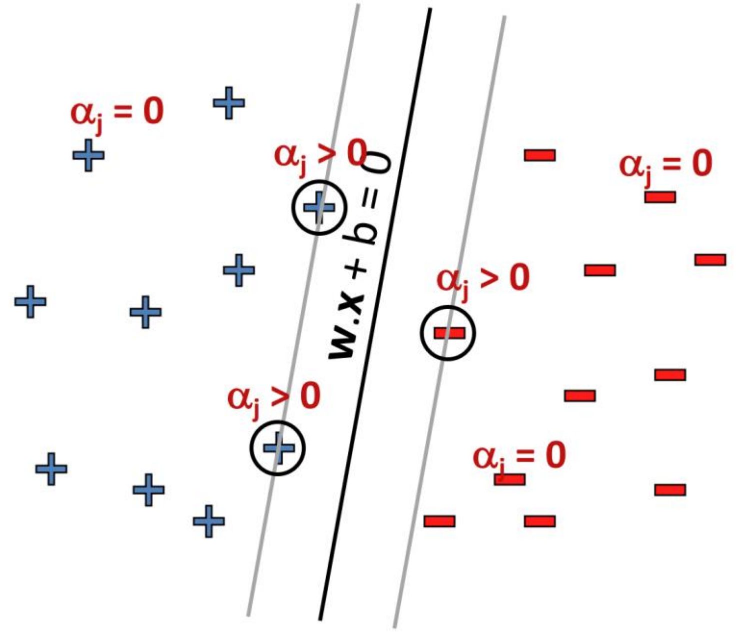
Hard-margin SVM: interpretation of dual variables

$$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

s.t. $\sum_i \alpha_i y_i = 0$

- Stationarity $\Rightarrow w^* = \sum_i \alpha_i^* y_i x_i$
- Support vectors: those data points i with $\alpha_i^* > 0$.
- Complementary slackness $\Rightarrow \alpha_i^* \left(1 - y_i\left({w^*}^\top x_i + b^*\right)\right) = 0$

i.e. $\alpha_i^* > 0 \Rightarrow y_i\left({w^*}^\top x_i + b^*\right) = 1$



- Implications:
  - Can use this to recover b^* from α^*
  - SVM “compresses” training set

https://www.cs.cmu.edu/~aarti/Class/10315_Fall20/lecs/svm_dual_kernel.pdf

The dual problem for soft-margin SVM

$$\min_{w,b,\xi_{1:n}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ \ y_i(w^\top x_i + b) \geq 1 - \xi_i, \forall i$$
$$\xi_i \geq 0, \forall i$$

- Lagrangian: $\mathcal{L}(w, b, \xi, \alpha, \gamma) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^\top x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\gamma_i\xi_i$
- Dual problem: maximize $D(\alpha, \gamma) := \min_{w,b,\xi} \mathcal{L}(w, b, \xi, \alpha, \gamma)$
- $\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i \cdot y_i x_i$
- $\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$
- $\frac{\partial \mathcal{L}}{\partial \xi_i} = C - (\alpha_i + \gamma_i) = 0$

The dual problem for soft-margin SVM (cont'd)

- Plugging the optimality conditions into $D(\alpha, \gamma) := \min_{w,b,\xi} \mathcal{L}(w, b, \xi, \alpha, \gamma)$, with some algebra, we have:

$$D(\alpha, \gamma) = \begin{cases} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j \,, & \sum_i \alpha_i y_i = 0 \,, \alpha_i + \gamma_i = C, \forall i \\ -\infty, & \text{otherwise} \end{cases}$$

- Dual problem: $\max_{\alpha \geq 0, \gamma \geq 0} D(\alpha, \gamma)$

- Representing γ in terms of α, the dual problem is equivalent to:

$$\max_{0 \leq \alpha \leq C} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

s.t. $\sum_i \alpha_i y_i = 0$

- Remark: for homogeneous version, same dual problem without equality constraint (exercise)

Soft-margin SVM: Support vectors

$$\max_{0 \le \alpha \le C} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

s.t. $\sum_i \alpha_i y_i = 0$

- Support vectors: those data points i with $\alpha_i^* > 0$.
- Stationary condition:
- $\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w^* = \sum_{i=1}^{n} \alpha_i^* \cdot y_i x_i$

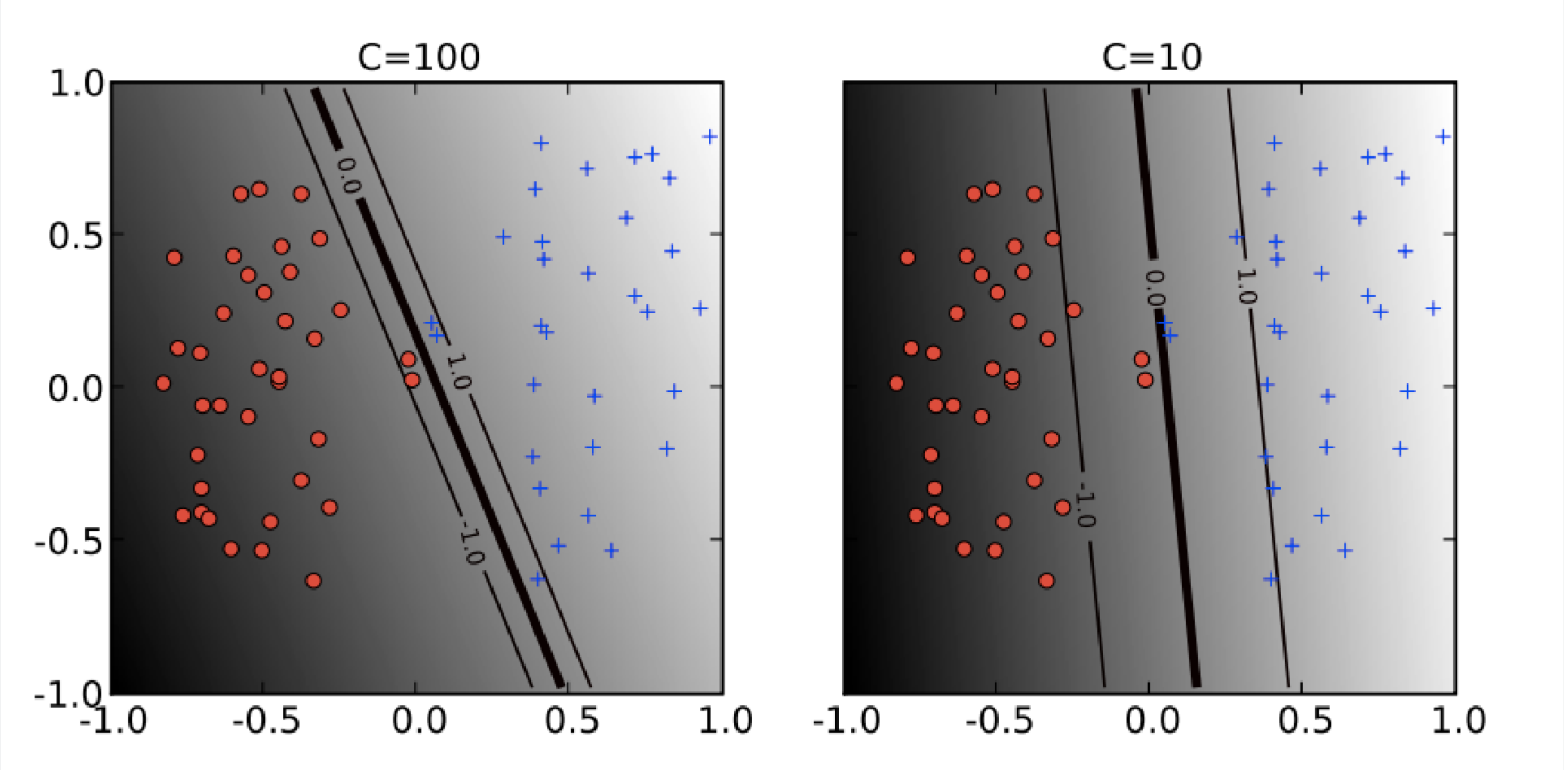
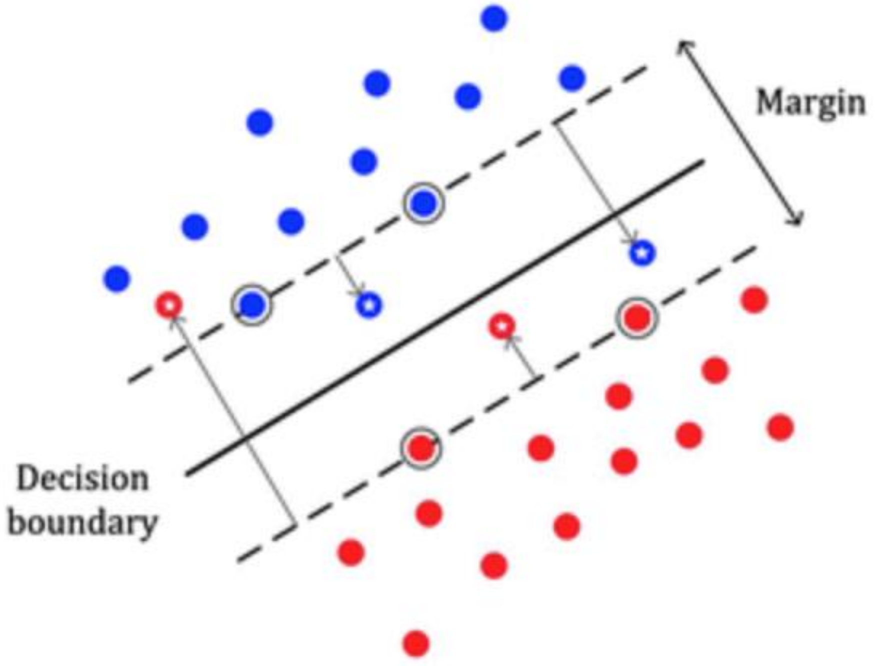


Figure from Ben-Hur & Weston, *Methods in Molecular Biology* 2010

Soft-margin SVM: additional remarks

- Complementary slackness ⇒

For all i, $\gamma_i^* \xi_i^* = 0$ and $\alpha_i^* \left(y_i \left({w^*}^\top x_i + b^* \right) - 1 + \xi_i^* \right) = 0$

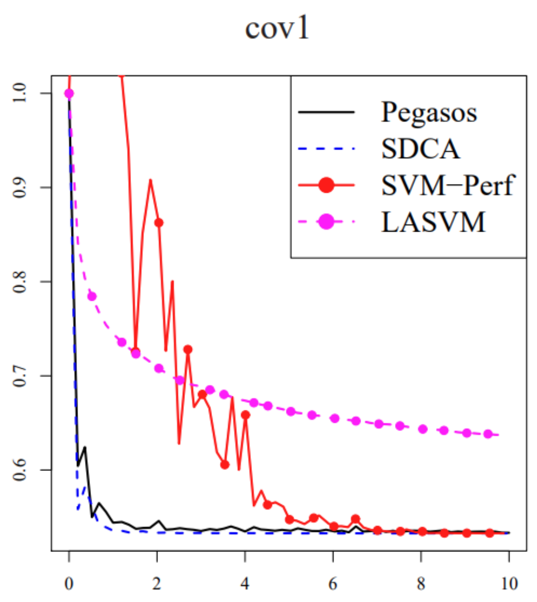


- Therefore, $\alpha_i^* > 0 \ \Rightarrow y_i \left({w^*}^\top x_i + b^* \right) = 1 - \xi_i^* \leq 1$

- $\alpha_i^* \in (0, C) \Rightarrow \gamma_i^* \in (0, C) \Rightarrow \xi_i^* = 0 \Rightarrow y_i \left({w^*}^\top x_i + b^* \right) = 1$
  - Use this to recover b^*

https://ankitnitjsr13.medium.com/math-behind-svm-support-vector-machine-864e58977fdb

Dual SVM: optimization



- Solving

$$\max_{0 \le \alpha \le C} D(\alpha) := \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

- In practice: use stochastic dual coordinate ascent (SDCA):
- For $t = 1,2,..$
  - Choose $i \sim \text{Uniform}(\{1, \ldots, n\})$
  - $\alpha_i \leftarrow \text{argmax}_{\alpha_i \in [0,C]} D(\alpha_1, \ldots, \alpha_i, \ldots, \alpha_n)$ – a univariate constrained quadratic maximization

- For the nonhomogeneous version:

$$\max_{0 \le \alpha \le C} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

s.t. $\sum_i \alpha_i y_i = 0$

- Popular algorithm: Sequential minimal optimization (SMO) (Platt, 1998)

SVM: summary

- Hinge loss & geometric motivation
- Optimization: finding the ERM
- Lagrange multiplier
  - I will include a few homework problems on this
- Dual formulation
  - why bother? kernel methods!

Next class (9/28)

- Kernel methods
- Assigned reading: CIML 11.4, 11.5 (Review of SVM dual formulation)