# Hyperplane based Classifiers (2): Large-Margin Classification - SVM

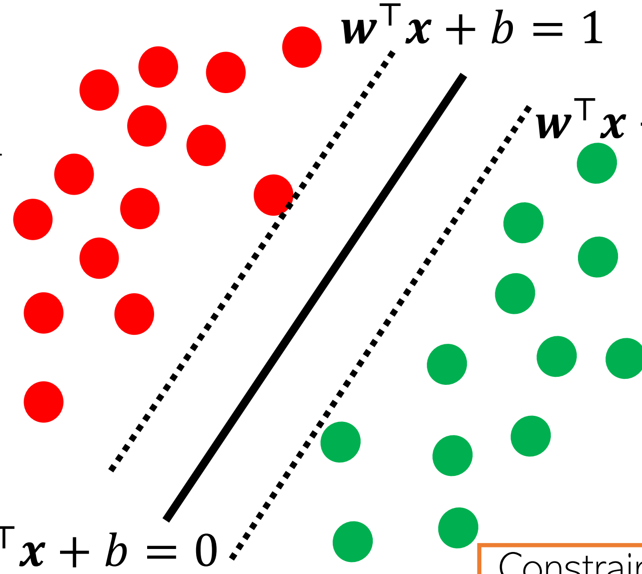
CS771: Introduction to Machine Learning

Piyush Rai

# Support Vector Machine (SVM)

SVM originally proposed by Vapnik and colleagues in early 90s

- Hyperplane based classifier. Ensures a large margin around the hyperplane
- Will assume a linear hyperplane to be of the form $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ (nonlinear ext. later)




$$\boldsymbol{w}^\top \boldsymbol{x}_n + b \geq 1 \quad \text{if} \quad y_n = +1$$

$$\boldsymbol{w}^\top \boldsymbol{x}_n + b \leq -1 \quad \text{if} \quad y_n = -1$$

$$y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1 \quad \forall n$$

Distance from the closest point (on either side)

"Margin" of the hyperplane

Distance of an input $\boldsymbol{x}_n$ from the h.p.

$$\gamma = \min_{1 \leq n \leq N} \frac{|\boldsymbol{w}^\top \boldsymbol{x}_n + b|}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}$$

Want the hyperplane $(\boldsymbol{w}, b)$ such that this margin is maximized (max-margin hyperplane) and $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1 \quad \forall n$

Constrained optimization problem

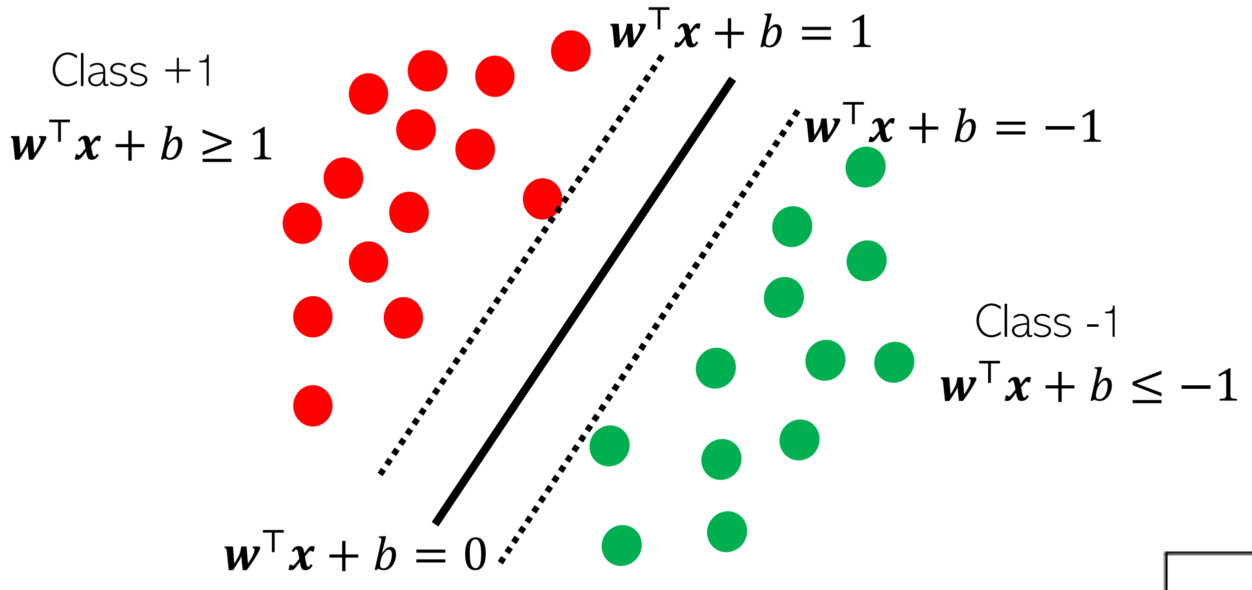
$$\text{Total margin} = \frac{2}{||\boldsymbol{w}||}$$

- Two other "supporting" hyperplanes defining a "no man's land"
  - Ensure that zero training examples fall in this region (will relax later)
  - The SVM idea: Position the hyperplane s.t. this region is as "wide" as possible

The 1/-1 in supp. h.p. equations is arbitrary; can replace by any scalar m/-m and solution won't change, except a simple scaling of $\boldsymbol{w}$

# Hard-Margin SVM

- Hard-Margin: Every training example must fulfil margin condition $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1$
- Meaning: Must not have any example in the no-man's land

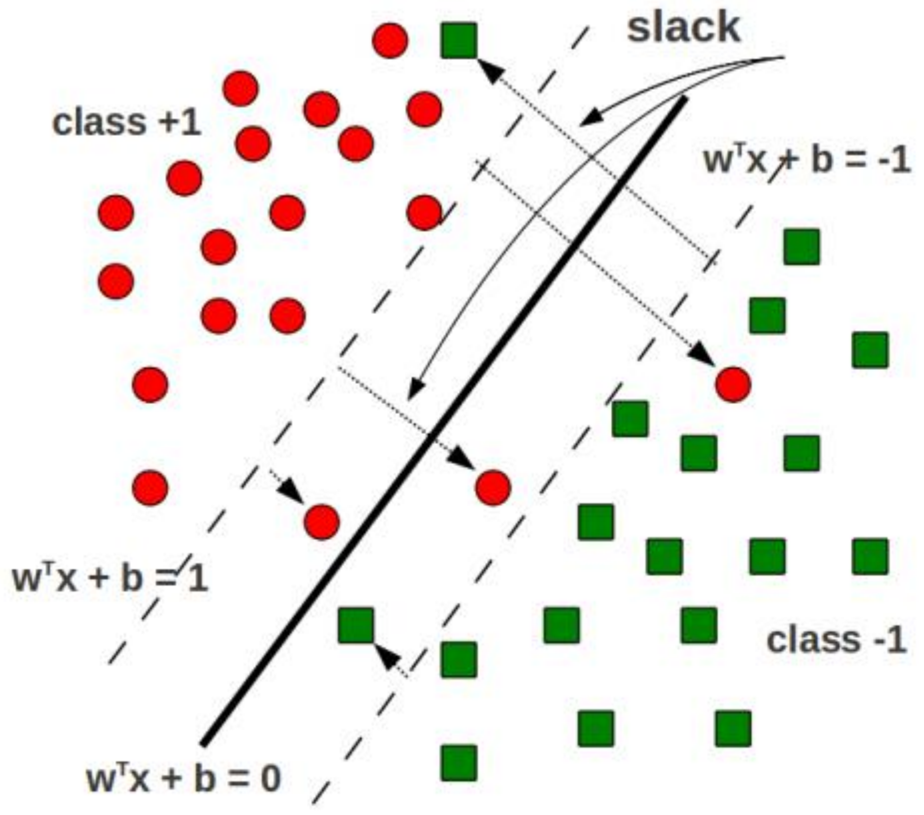



- Also want to maximize margin $2\gamma = \frac{2}{||\boldsymbol{w}||}$
- Equivalent to <u>minimizing</u> $||\boldsymbol{w}||^2$ or $\frac{||\boldsymbol{w}||^2}{2}$
- The objective func. for hard-margin SVM

$$\min_{\boldsymbol{w}, b} \quad f(\boldsymbol{w}, b) = \frac{||\boldsymbol{w}||^2}{2}$$
$$\text{subject to} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geq 1, \qquad n = 1, \ldots, N$$

Constrained optimization problem with $N$ inequality constraints. Objective and constraints both are convex

# Soft-Margin SVM (More Commonly Used)




- Allow some training examples to fall within the no-man's land (margin region)
- Even okay for some training examples to fall totally on the wrong side of h.p.
- Extent of "violation" by a training input $(\boldsymbol{x}_n, y_n)$ is known as slack $\xi_n \geq 0$
- $\xi_n > 1$ means totally on the wrong side

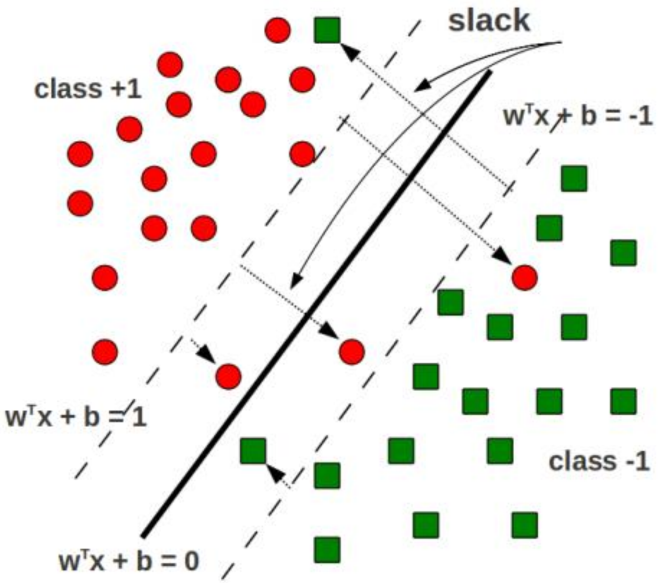
$$\boldsymbol{w}^\top \boldsymbol{x}_n + b \geq 1 - \xi_n \quad \text{if} \quad y_n = +1$$
$$\boldsymbol{w}^\top \boldsymbol{x}_n + b \leq -1 + \xi_n \quad \text{if} \quad y_n = -1$$

Soft-margin constraint: $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1 - \xi_n \quad \forall n$

# Soft-Margin SVM (Contd)

- Goal: Still want to maximize the margin such that
  - Soft-margin constraints $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1 - \xi_n$ are satisfied for all training ex.
  - Do not have too many margin violations (sum of slacks $\sum_{n=1}^{N} \xi_n$ should be small)

Sum of slacks is like the training error




- The objective func. for soft-margin SVM

Inversely prop. to margin

Trade-off hyperparam

training error

Constrained optimization problem with $2N$ inequality constraints. Objective and constraints both are convex

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} f(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{||\boldsymbol{w}||^2}{2} + C \sum_{n=1}^{N} \xi_n$$

$$\text{subject to} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0 \quad n = 1, \ldots, N$$

- Hyperparameter $C$ controls the trade off between large margin and small training error (need to tune)
  - Large $C$: small training error but also small margin (bad)
  - Small $C$: large margin but large training error (bad)

# Solving the SVM Problem

# Solving Hard-Margin SVM

- The hard-margin SVM optimization problem is

$$\min_{\boldsymbol{w},b} \quad f(\boldsymbol{w},b) = \frac{||\boldsymbol{w}||^2}{2}$$
$$\text{subject to} \quad 1 - y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \leq 0, \qquad n = 1,\ldots,N$$

- A constrained optimization problem. One option is to solve using Lagrange's method
- Introduce Lagrange multipliers $\alpha_n$ $(n = 1, \ldots, N)$, one for each constraint, and solve

$$\min_{\boldsymbol{w},b} \max_{\alpha \geq 0} \mathcal{L}(\boldsymbol{w},b,\alpha) = \frac{||\boldsymbol{w}||^2}{2} + \sum_{n=1}^{N} \alpha_n\{1 - y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b)\}$$

- $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ denotes the vector of Lagrange multipliers
- It is easier (and helpful; we will soon see why) to solve the dual: min and then max

# Solving Hard-Margin SVM

Note: if we ignore the bias term $b$ then we don't need to handle the constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$ (problem becomes a bit more easy to solve)

- The dual problem (min then max) is

$$\boxed{\max_{\boldsymbol{\alpha} \geq 0} \min_{\boldsymbol{w}, b} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{\boldsymbol{w}^\top \boldsymbol{w}}{2} + \sum_{n=1}^{N} \alpha_n \{1 - y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b)\}}$$

Otherwise, the $\alpha_n$'s are coupled and some opt. techniques such as co-ordinate ascent can't easily be applied

- Take (partial) derivatives of $\mathcal{L}$ w.r.t. $\boldsymbol{w}$ and $b$ and setting them to zero gives (verify)

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boxed{\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n} \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0$$

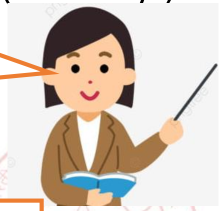
$\alpha_n$ tells us how important training example $(\boldsymbol{x}_n, y_n)$ is

- The solution $\boldsymbol{w}$ is simply a weighted sum of all the training inputs
- Substituting $\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n$ in the Lagrangian, we get the dual problem as (verify)

This is also a "quadratic program" (QP) – a quadratic function of the variables $\boldsymbol{\alpha}$

$$\boxed{\max_{\boldsymbol{\alpha} \geq 0} \mathcal{L}_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m,n=1}^{N} \alpha_m \alpha_n y_m y_n (\boldsymbol{x}_m^T \boldsymbol{x}_n)}$$

Note that inputs appear only as pairwise dot products. This will be useful later on when we make SVM nonlinear using kernel methods



Maximizing a concave function (or minimizing a convex function) s.t. $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$ . Many methods to solve it.

$$\boxed{\max_{\boldsymbol{\alpha} \geq 0} \mathcal{L}_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{G} \boldsymbol{\alpha}}$$

**G** is an $N \times N$ p.s.d. matrix, also called the Gram Matrix, $G_{nm} = y_n y_m \boldsymbol{x}_n^\top \boldsymbol{x}_m$, and **1** is a vector of all 1s

(Note: For various SVM solvers, can see "Support Vector Machine Solvers" by Bottou and Lin)

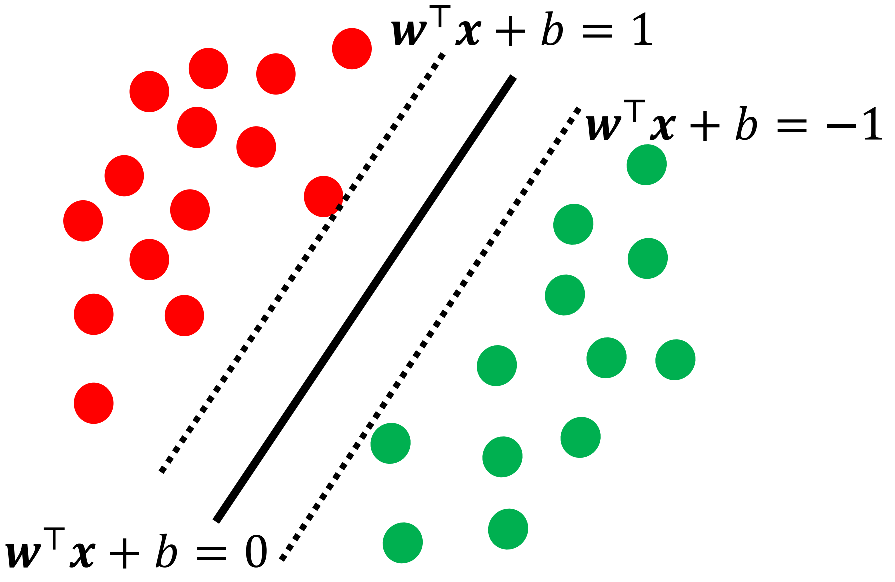
# Solving Hard-Margin SVM

- One we have the $\boldsymbol{\alpha_n}$'s by solving the dual, we can get $\boldsymbol{w}$ and $\boldsymbol{b}$ as

$$\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n \qquad \text{(we already saw this)}$$

$$b = -\frac{1}{2}\left(\min_{n:y_n=+1} \boldsymbol{w}^T \boldsymbol{x}_n + \max_{n:y_n=-1} \boldsymbol{w}^T \boldsymbol{x}_n\right) \qquad \text{(exercise)}$$

- A nice property: Most $\boldsymbol{\alpha_n}$'s in the solution will be zero (sparse solution)




- Reason: KKT conditions
- For the optimal $\boldsymbol{\alpha_n}$'s, we must have
- Thus $\boldsymbol{\alpha_n}$ nonzero only if $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) = 1$, i.e., the training example lies on the boundary

$$\alpha_n\{1 - y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b)\} = 0$$

- These examples are called support vectors

# Solving Soft-Margin SVM

- Recall the soft-margin SVM optimization problem

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad f(\boldsymbol{w},b,\boldsymbol{\xi}) = \frac{||\boldsymbol{w}||^2}{2} + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to} \quad 1 \leq y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) + \xi_n, \quad -\xi_n \leq 0 \qquad n = 1,\ldots,N$$

- Here $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_N]$ is the vector of slack variables
- Introduce Lagrange multipliers $\alpha_n, \beta_n$ for each constraint and solve Lagrangian

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \max_{\alpha\geq 0,\beta\geq 0} \mathcal{L}(\boldsymbol{w},b,\xi,\alpha,\beta) = \frac{||\boldsymbol{w}||^2}{2} + +C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n\{1 - y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) - \xi_n\} - \sum_{n=1}^{N}\beta_n\xi_n$$

- The terms in red color above were not present in the hard-margin SVM
- Two set of dual variables $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_N]$
- Will eliminate the primal var $\boldsymbol{w}$, $b$, $\boldsymbol{\xi}$ to get dual problem containing the dual variables

# Solving Soft-Margin SVM

Note: if we ignore the bias term $b$ then we don't need to handle the constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$ (problem becomes a bit more easy to solve)

Otherwise, the $\alpha_n$'s are coupled and some opt. techniques such as co-ordinate aspect can't easily applied

- The Lagrangian problem to solve

$$\boxed{\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \max_{\boldsymbol{\alpha}\geq 0,\boldsymbol{\beta}\geq 0} \mathcal{L}(\boldsymbol{w}, b, \xi, \alpha, \beta) = \frac{||\boldsymbol{w}||^2}{2} + +C\sum_{n=1}^{N} \xi_n + \sum_{n=1}^{N} \alpha_n\{1 - y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - \xi_n\} - \sum_{n=1}^{N} \beta_n \xi_n}$$

- Take (partial) derivatives of $\mathcal{L}$ w.r.t. $\boldsymbol{w}, b$, and $\xi_n$ and setting to zero gives

Weighted sum of training inputs

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boxed{\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n}, \quad \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0, \quad \frac{\partial \mathcal{L}}{\partial \xi_n} = 0 \Rightarrow C - \alpha_n - \beta_n = 0$$

- Using $C - \alpha_n - \beta_n = 0$ and $\beta_n \geq 0$, we have $\alpha_n \leq C$ (for hard-margin, $\alpha_n \geq 0$)
- Substituting these in the Lagrangian $\mathcal{L}$ gives the Dual problem

The dual variables $\boldsymbol{\beta}$ don't appear in the dual problem!

Given $\boldsymbol{\alpha}$, $\boldsymbol{w}$ and $b$ can be found just like the hard-margin SVM case

$$\boxed{\max_{\boldsymbol{\alpha}\leq C,\boldsymbol{\beta}\geq 0} \mathcal{L}_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{m,n=1}^{N} \alpha_m \alpha_n y_m y_n (\boldsymbol{x}_m^T \boldsymbol{x}_n) \quad \text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0}$$

Maximizing a concave function (or minimizing a convex function) s.t. $\boldsymbol{\alpha} \leq \boldsymbol{C}$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$ . Many methods to solve it.

$$\boxed{\max_{\boldsymbol{\alpha}\leq C} \mathcal{L}_D(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{G}\boldsymbol{\alpha}}$$

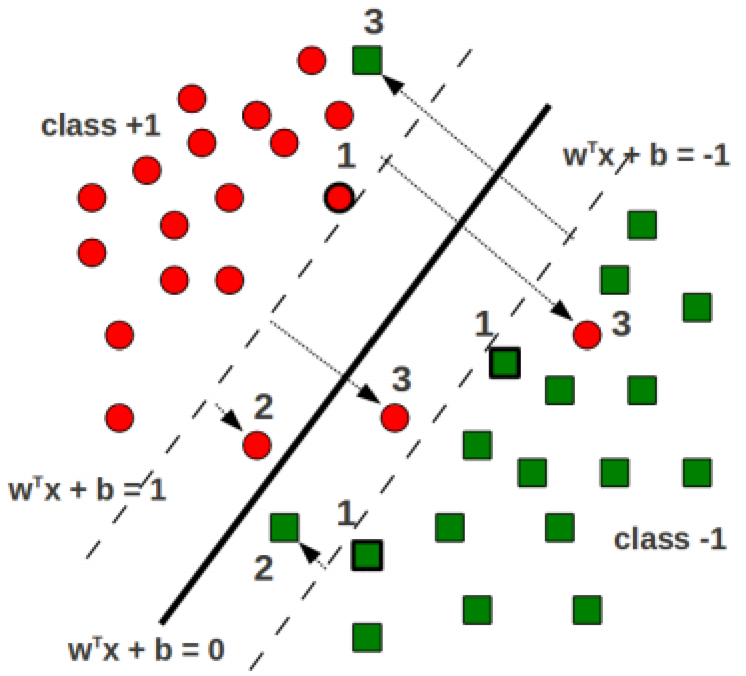
In the solution, $\boldsymbol{\alpha}$ will still be sparse just like the hard-margin SVM case. Nonzero $\alpha_n$ correspond to the support vectors

(Note: For various SVM solvers, can see "Support Vector Machine Solvers" by Bottou and Lin)

# Support Vectors in Soft-Margin SVM

- The hard-margin SVM solution had only one type of support vectors
  - All lied on the supporting hyperplanes $\boldsymbol{w}^\top \boldsymbol{x}_n + b = 1$ and $\boldsymbol{w}^\top \boldsymbol{x}_n + b = -1$

- The soft-margin SVM solution has <u>three</u> types of support vectors (with nonzero $\alpha_n$)




1. Lying on the supporting hyperplanes
2. Lying within the margin region but still on the correct side of the hyperplane
3. Lying on the wrong side of the hyperplane (misclassified training examples)

# SVMs via Dual Formulation: Some Comments

- Recall the final dual objectives for hard-margin and soft-margin SVM

Hard-Margin SVM: $\max_{\boldsymbol{\alpha} \geq 0} \mathcal{L}_D(\alpha) = \boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{G}\boldsymbol{\alpha}$

Soft-Margin SVM: $\max_{\boldsymbol{\alpha} \leq C} \mathcal{L}_D(\alpha) = \boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{G}\boldsymbol{\alpha}$

Note: Both these ignore the bias term $b$ otherwise will need another constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$

- The dual formulation is nice due to two primary reasons
  - Allows conveniently handling the margin based constraint (via Lagrangians)
  - Allows learning nonlinear separators by replacing inner products in $G_{nm} = y_n y_m \boldsymbol{x}_n^\top \boldsymbol{x}_m$ by general kernel-based similarities (more on this when we talk about kernels)
- However, dual formulation can be expensive if $N$ is large (esp. compared to $D$)
  - Need to solve for $N$ variables $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$
  - Need to pre-compute and store $N \times N$ gram matrix $\mathbf{G}$
- Lot of work on speeding up SVM in these settings (e.g., can use co-ord. descent for $\boldsymbol{\alpha}$)

# Solving for SVM in the Primal

- Maximizing margin subject to constraints led to the soft-margin formulation of SVM

$$\underset{\boldsymbol{w},b,\xi}{\arg\min} \frac{||\boldsymbol{w}||^2}{2} + C\sum_{n=1}^{N} \xi_n$$

$$\text{subject to} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0 \qquad n = 1, \ldots, N$$

- Note that slack $\xi_n$ is the same as $\max\{0, 1 - y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b)\}$, i.e., hinge loss for $(\boldsymbol{x}_n, y_n)$
- Thus the above is equivalent to minimizing the $\ell_2$ regularized hinge loss

$$\mathcal{L}(\boldsymbol{w}, b) = \sum_{n=1}^{N} \max\{0, 1 - y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b)\} + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

- Sum of slacks is like sum of hinge losses, $C$ and $\lambda$ play similar roles
- Can learn $(\boldsymbol{w}, b)$ directly by minimizing $\mathcal{L}(\boldsymbol{w}, b)$ using (stochastic) (sub)grad. descent
  - Hinge-loss version preferred for linear SVMs, or with other regularizers on $\boldsymbol{w}$ (e.g., $\ell_1$)

# SVM: Summary

- A hugely (perhaps the most!) popular classification algorithm
- Reasonably mature, highly optimized SVM softwares freely available (perhaps the reason why it is more popular than various other competing algorithms)
- Some popular ones: libSVM, LIBLINEAR, sklearn also provides SVM
- Lots of work on scaling up SVMs* (both large $N$ and large $D$)
- Extensions beyond binary classification (e.g., multiclass, structured outputs)
- Can even be used for regression problems (Support Vector Regression)
- Nonlinear extensions possible via kernels

* See: "Support Vector Machine Solvers" by Bottou and Lin

# Coming up next

- A co-ordinate ascent algorithm for solving the SVM dual
- Multi-class SVM
- One-class SVM
- Kernel methods and nonlinear SVM via kernels