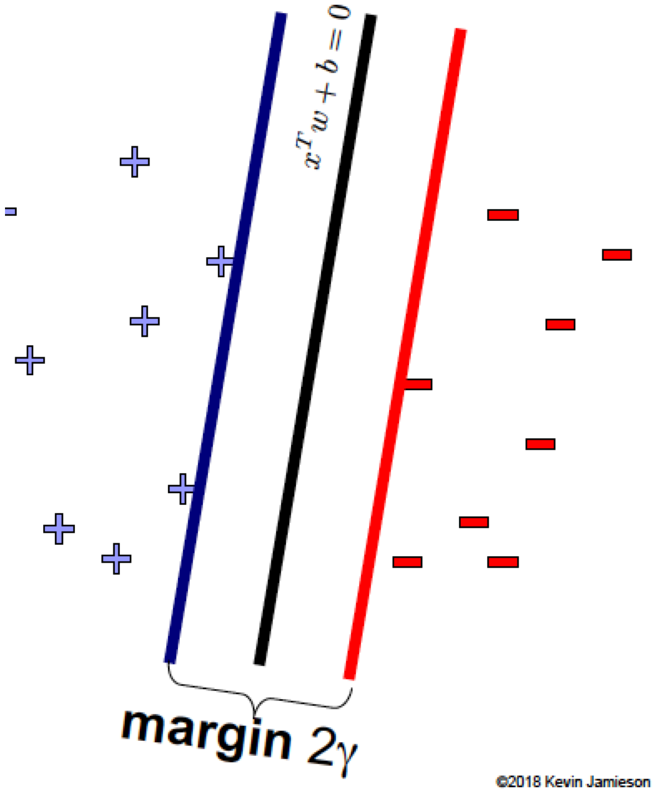
Support Vector Machines

CSE 446: Machine Learning
Slides by Emily Fox + Kevin Jamieson + others
Presented by Anna Karlin

May 1, 2019

Linear classifiers—Which line is better?


$x^T w + b = 0$
margin 2γ
©2018 Kevin Jamieson

Maximizing the margin for linearly separable data

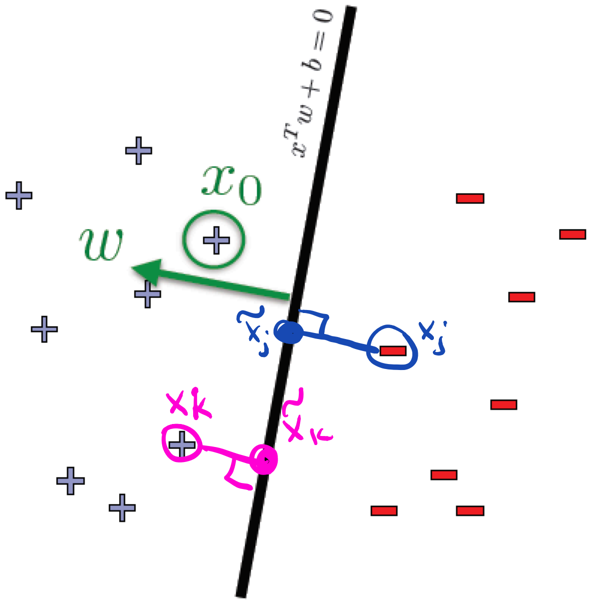
$\frac{w}{\|w\|}$ unit vector

$u \cdot v = \|u\| \, \|v\| \cos\theta$

$\tilde{x}_k \cdot w + b = 0$

$\tilde{x}_j \cdot w + b = 0$

Given hyperplane, what is margin?



Distance from x_0 to hyperplane defined by $x^T w + b = 0$?

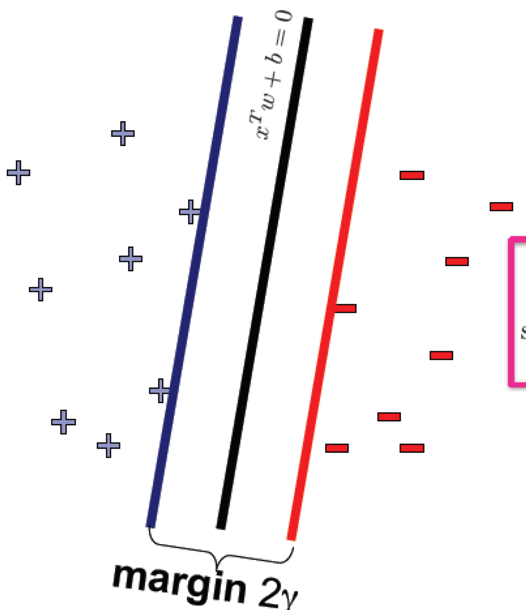
$y_k = +1$ $\quad \|x_k - \tilde{x}_k\| = (x_k - \tilde{x}_k) \cdot \frac{w}{\|w\|} = \frac{x_k \cdot w + b}{\|w\|} \geq \gamma$

$y_j = -1$ $\quad (x_j - \tilde{x}_j) \cdot \frac{w}{\|w\|} = -\|x_j - \tilde{x}_j\|$

want $\quad -\frac{(x_j \cdot w + b)}{\|w\|} \geq \gamma$

want all the +1's to be on the side that w is pointing towards and all the -1's on the opposite side.

Our optimization problem



constrained optimization problem.

Optimal Hyperplane

$$\boxed{\begin{aligned}&\max_{w,b} \gamma \\ &\text{subject to } \frac{1}{\|w\|_2} y_i(x_i^T w + b) \geq \gamma \quad \forall i\end{aligned}}$$

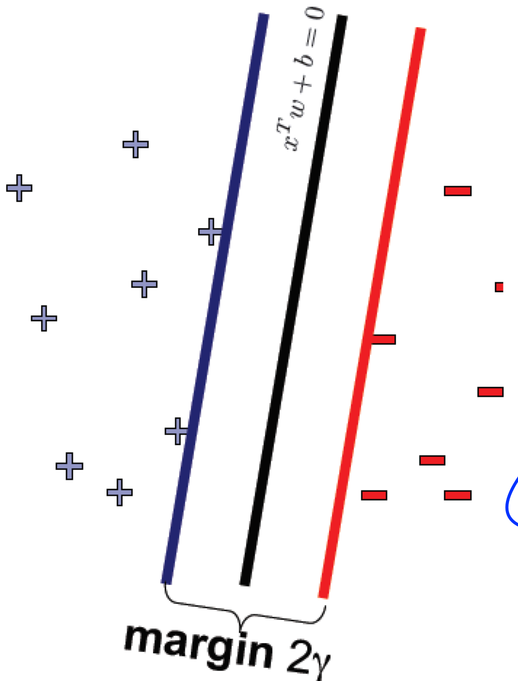
inequality is invariant to scaling. Why optimize something so complicated when we can simplify it to be easy to work with

replace w,b with 100w and 100b, the normalizing constant of 1/ llwll on the LHS will give us the same inequality. Why not choose a scaling that gives a convenient formalization?

Choose such that gamma * llwll = 1

then gamma = 1/ llwll   and we can remove the gamma variable entirely!

Final version



Solvable efficiently –
quadratic programming problem

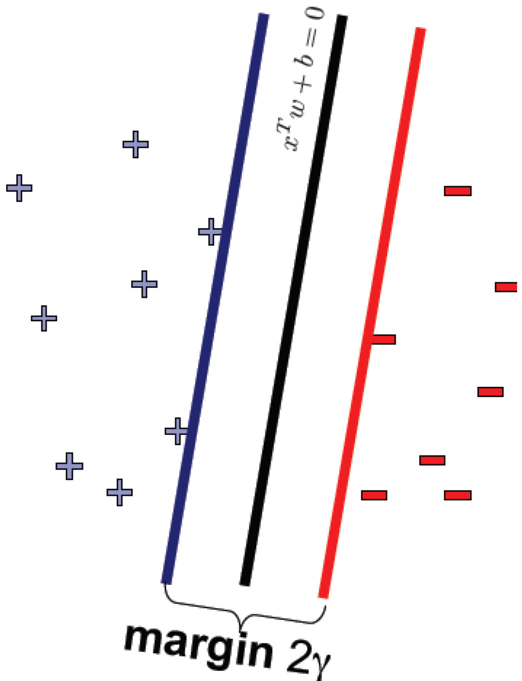
can be solved efficiently

changed to a minimization problem.

Optimal Hyperplane (reparameterized)

$$\min_{w,b} ||w||_2^2$$
$$\text{subject to } y_i(x_i^T w + b) \geq 1 \quad \forall i$$

What are support vectors?



Optimal Hyperplane (reparameterized)

$$\min_{w,b} ||w||_2^2$$

$$\text{subject to } y_i(x_i^T w + b) \geq 1 \quad \forall i$$

22

What if the data are not linearly separable?

What if data are not linearly separable?

Use feature maps…

increase dimensionality of the data:

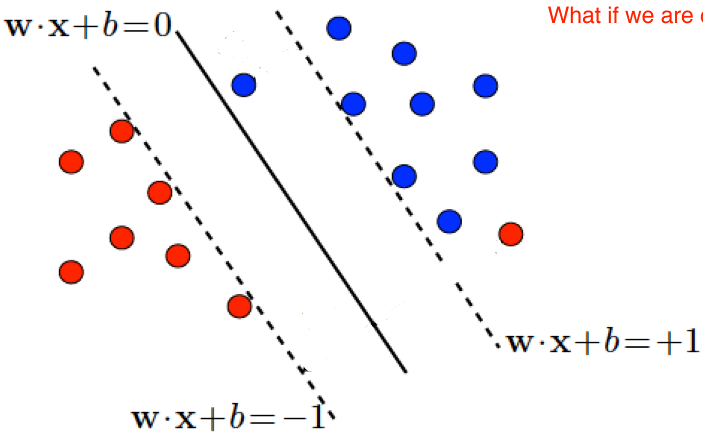
(x,y) —> (x,y, x^2, y^2, sin(x), cos(x)) etc.

SVMs can be kernelized!!!!

see the kernel trick from previous lectures.

All computations can be written as inner products

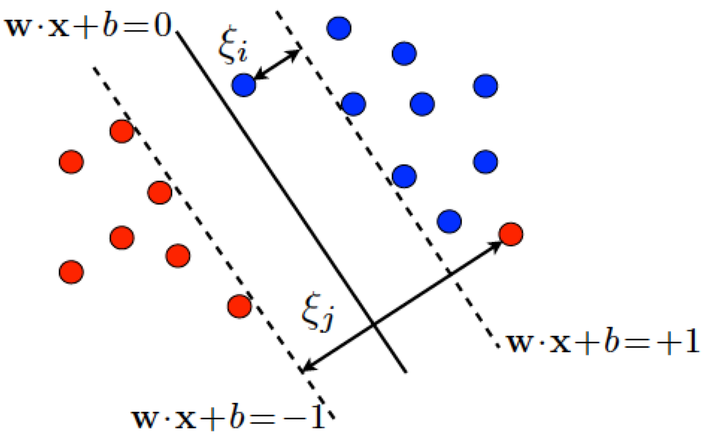
What if data are still not linearly separable?



might not be possible to get ALL data points on the right side of the separating hyperplane
What if we are ok with making a few mistakes, but otherwise get a large margin?

Courtesy Mehryar Mohri

What if data are still not linearly separable?

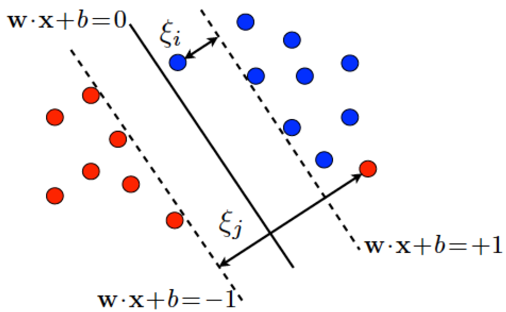


Courtesy Mehryar Mohri

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i$$

slack: distance to the "correct" side for each misclassified data point.

What if data are still not linearly separable?



Courtesy Mehryar Mohri

$$\min_{\mathbf{w},b,\xi} ||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i.$$

NOT considering positive slack. i.e. does not matter how far each point is into “correct” territory, only how far each wrongly classified point is from correct side.

non-negative constant C = hyperparameter; how much do I care about large margin vs low slack?
increase C = dont want slack, decrease = want large margin

fix w, b

suppose y_j (wx_j + b) >= 1 already correctly classified so we do not need any slack.
Choose E_j = 0

Suppose y_j (wx_j + b) < 1 i.e. incorrectly classified so we need some slack. The smallest slack we can add back is the difference
Choose E_j = 1-y_j (wx_j + b)

Thus we always know what E_j is! we can rewrite our objective function

min llwll^2 + C* sum_i of (max(0, 1-y_i (wx_i + b))

SOFT SVM

Final objective

min over w,b

max(0, 1-y_i(wx_i + b)

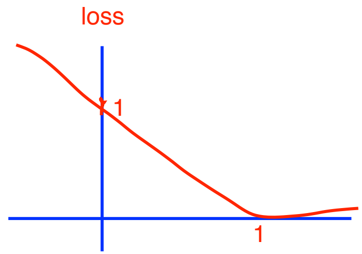
$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \lambda||\mathbf{w}||_2^2$$

loss on ith data point + regularization

Let lambda = 1/(nC)

Gradient descent for SVMs

Hinge loss



y(wx + b) = prediction

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i((\mathbf{w}^T\mathbf{x}_i + b))_+ + \lambda||\mathbf{w}||_2^2$$

- Hinge loss: $\ell((\mathbf{x}, y), \mathbf{w}) = (1 - y((\mathbf{w}^T\mathbf{x} + b))_+$

- Subgradient of hinge loss:

subgradient with respect to w
dL_w =

-yx if y(wx + b) < 1
[-yx, 1] if y(wx + b) = 1
0 if y(wx + b) > 1

Subgradient descent for hinge minimization

- Given data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
- Want to minimize:

$$\frac{1}{n}\sum_{i=1}^{n} \ell((\mathbf{x}_i, y_i), \mathbf{w}) + \lambda||\mathbf{w}||_2^2 = \frac{1}{n}\sum_{i}(1 - y_i((\mathbf{w}^T\mathbf{x}_i + b))_+ + \lambda||\mathbf{w}||_2^2$$

- As we've discussed, subgradient descent works like gradient descent:
  - But if there are multiple subgradients at a point, just pick (any) one:

$$\partial_{\mathbf{w}}\ell((\mathbf{x}, y), \mathbf{w}) = \mathbb{I}\{y(\mathbf{w}^T\mathbf{x} + b) \leq 1\}(-y\mathbf{x})$$

indicator random variable

Subgradient descent for hinge minimization

- Given data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

- Want to minimize:

$$\frac{1}{n}\sum_{i=1}^{n} \ell((\mathbf{x}_i, y_i), \mathbf{w}) + \lambda||\mathbf{w}||_2^2 = \frac{1}{n}\sum_i (1 - y_i((\mathbf{w}^T\mathbf{x}_i + b))_+ + \lambda||\mathbf{w}||_2^2$$

- As we′ve discussed, subgradient descent works like gradient descent:
  - But if there are multiple subgradients at a point, just pick (any) one:

$$\begin{aligned}
\mathbf{w}_{t+1} &:= \mathbf{w}_t - \eta\left(\frac{1}{n}\sum_{i=1}^{n} \partial_{\mathbf{w}}\ell((\mathbf{x}_i, y_i), \mathbf{w}) + 2\lambda\mathbf{w}_t\right) \\
&= \mathbf{w}_t - \eta\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y(\mathbf{w}_t \cdot \mathbf{x}_i + b) \leq 1\}(-y_i\mathbf{x}_i) + 2\lambda\mathbf{w}_t\right) \\
&= \mathbf{w}_t + \eta\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y(\mathbf{w}_t \cdot \mathbf{x}_i + b) \leq 1\}(y_i\mathbf{x}_i) - \eta 2\lambda\mathbf{w}_t.
\end{aligned}$$

$$\partial_{\mathbf{w}}\ell((\mathbf{x}, y), \mathbf{w}) = \mathbb{I}\{y(\mathbf{w}^T\mathbf{x} + b) \leq 1\}(-y\mathbf{x})$$

SVM

- **(Sub)gradient Descent Update**

$$
\begin{aligned}
\mathbf{w}_{t+1} &:= \mathbf{w}_t - \eta\left(\frac{1}{n}\sum_{i=1}^{n} \partial_{\mathbf{w}} \ell((\mathbf{x}_i, y_i), \mathbf{w}) + 2\lambda \mathbf{w}_t\right) \\
&= \mathbf{w}_t - \eta\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y(\mathbf{w}_t \cdot \mathbf{x}_i + b) \leq 1\}(-y_i \mathbf{x}_i) + 2\lambda \mathbf{w}_t\right) \\
&= \mathbf{w}_t + \eta \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y(\mathbf{w}_t \cdot \mathbf{x}_i + b) \leq 1\}(y_i \mathbf{x}_i) - \eta 2\lambda \mathbf{w}_t.
\end{aligned}
$$

since we have the 1/n our losses are normalized. Thus, in SGD we do not have to worry about scaling the regularizer

- **SGD update**

$$
\mathbf{w}_{t+1} := \mathbf{w}_t + \eta\, \mathbb{I}\{y(\mathbf{w}_t \cdot \mathbf{x}_i + b) \leq 1\}(y_i \mathbf{x}_i) - \eta 2\lambda \mathbf{w}_t.
$$

Machine learning problems

- Given i.i.d. data set:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$$

- Find parameters **w** to minimize average loss

(or regularized version):

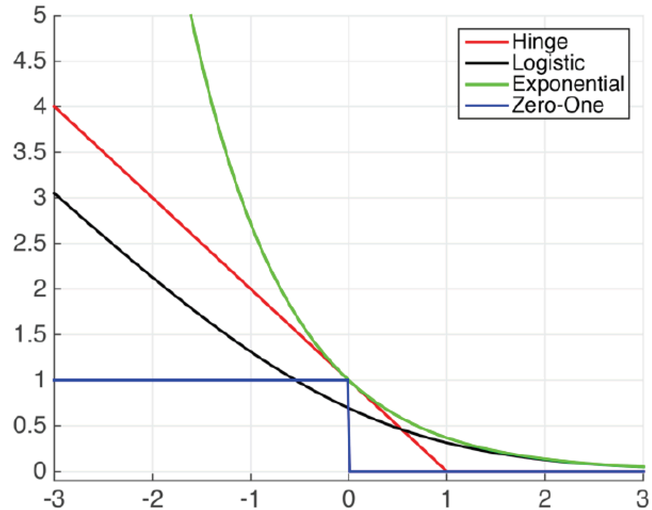
$$\frac{1}{n}\sum_{i=1}^{n} \ell_i(\mathbf{w})$$

Squared loss: $\ell_i(\mathbf{w}) = (y_i - x_i^T \mathbf{w})^2$

Logistic loss: $\ell_i(\mathbf{w}) = \log(1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w}))$

Hinge loss: $\ell_i(\mathbf{w}) = \max\{0, 1 - y_i \mathbf{x}_i^T \mathbf{w}\}$

Courtesy Killian Weinberger



What you need to know…

- Maximizing margin
- Derivation of SVM formulation
- Non-linearly separable case
  - Hinge loss
  - a.k.a. adding slack variables
- Can optimize SVMs with SGD
  - Many other approaches possible