Support Vector Machine (SVM)

Slides adapted from Nicholas Ruozzi, David Sontag and Vibhav Gogate

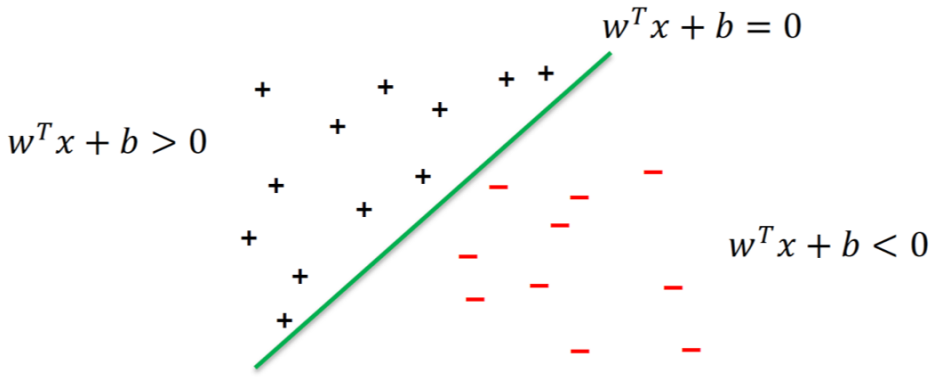
What is a good classifier?

Labeled Data

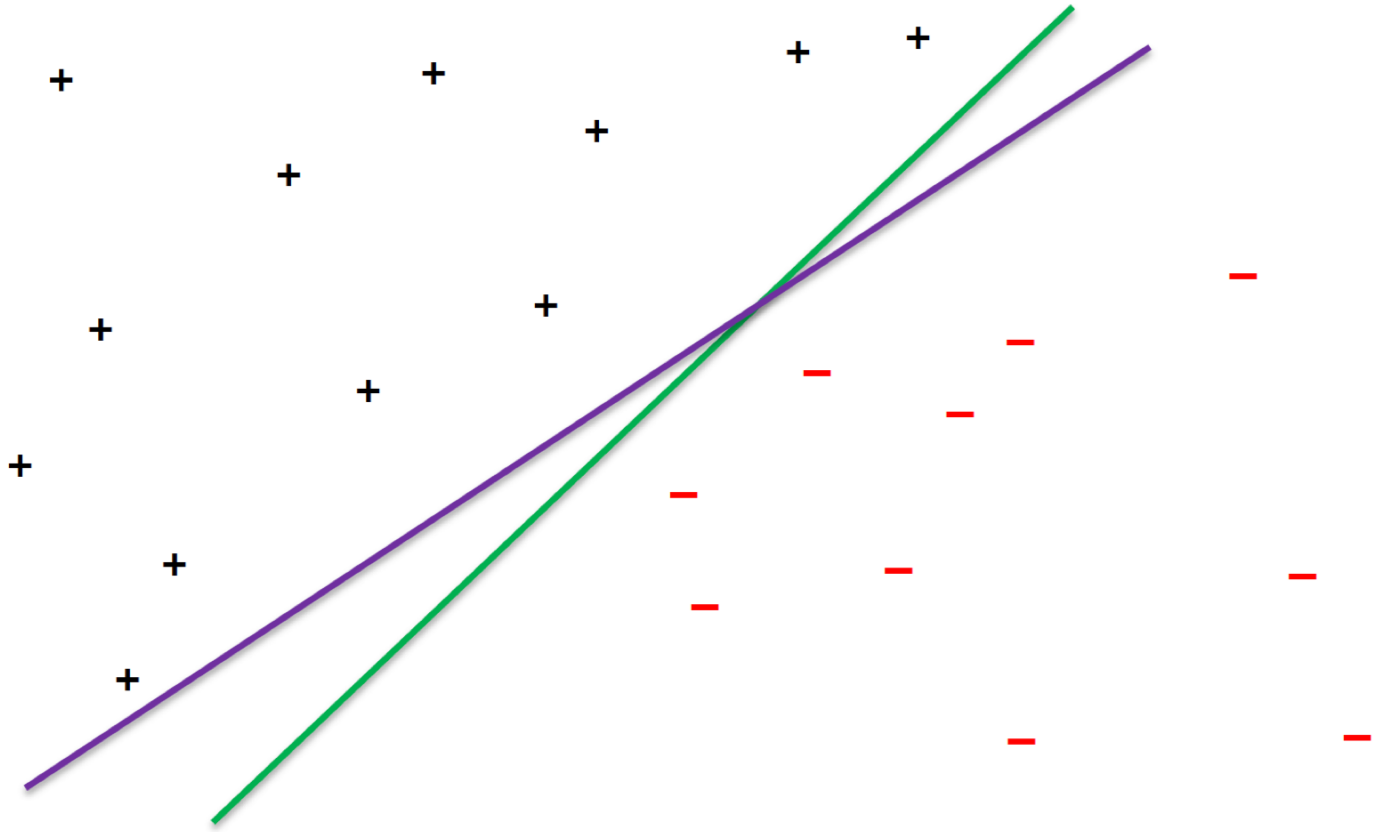
$(x^{(1)},y^{(1)})$, …, $(x^{(M)},y^{(M)})$ with $x^{(i)}\in\mathbb{R}^{(n)}$ and $y^{(i)}\in\{-1,+1\}$

Classifier

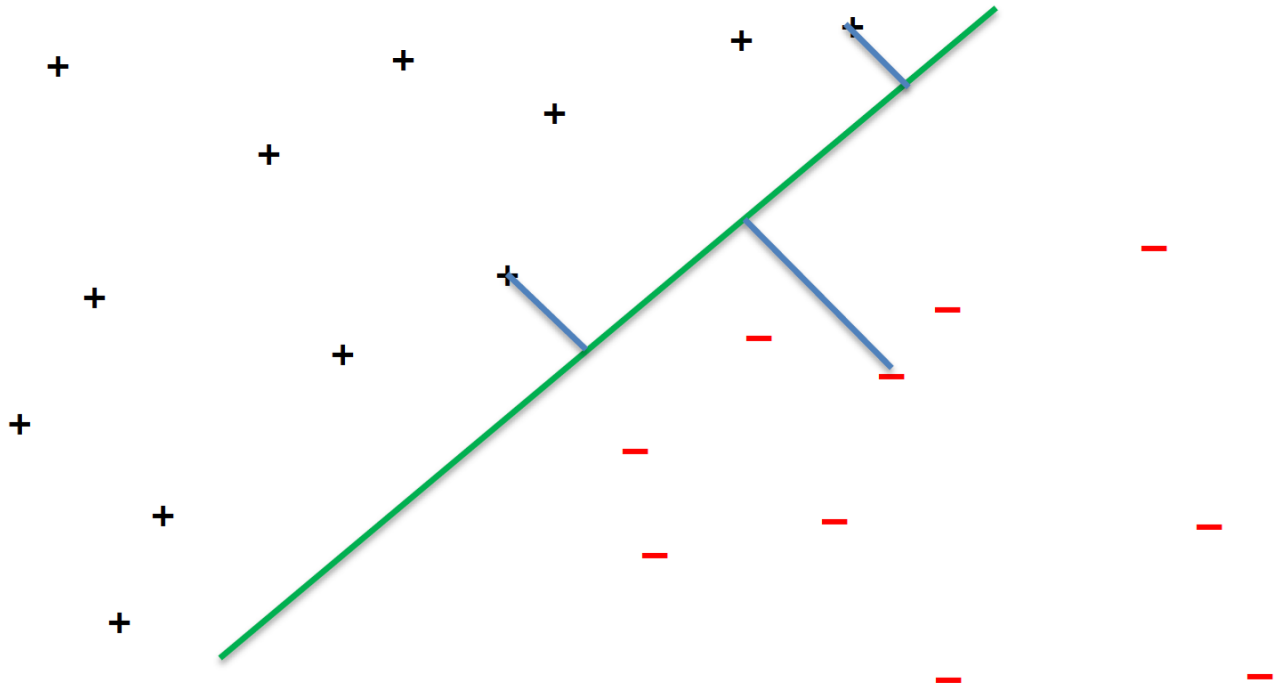
A separator that puts + and – data points on opposite sides



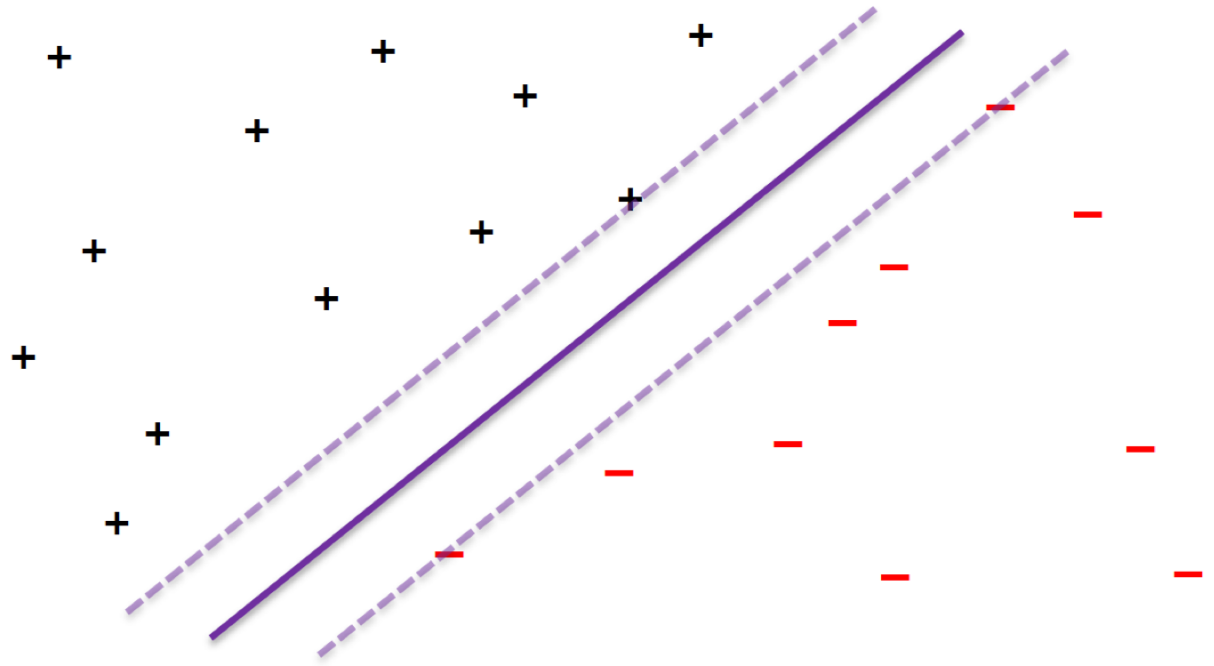
What is a good classifier?



Define the margin to be the distance of the closest data point to the classifier

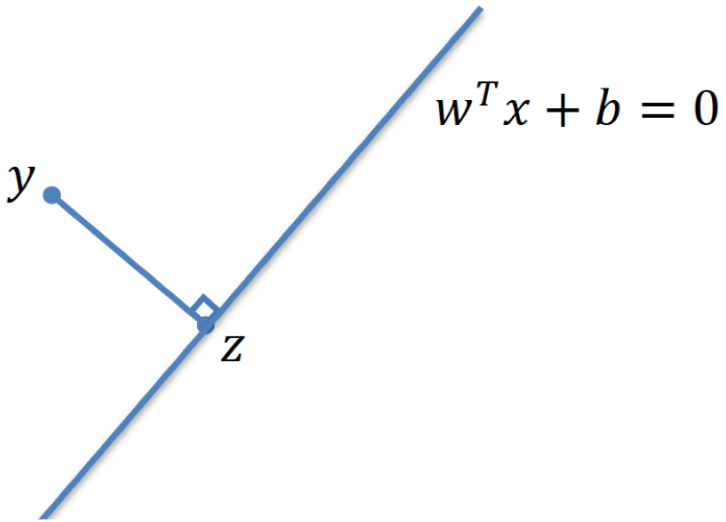


SVM: find the classifier that maximize the margin



How to find out the 'perfect' classifier?

Mathematical definition



$$y - z = \|y - z\| \frac{w}{\|w\|}$$

See derivation on whiteboard

- This analysis yields the following optimization problem

$$\max_{w} \frac{1}{\|w\|}$$

such that

$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \text{for all } i$$

- Or, equivalently,

$$\min_{w} \|w\|^2$$

such that

$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \text{for all } i$$

$$\min_{w} \|w\|^2$$

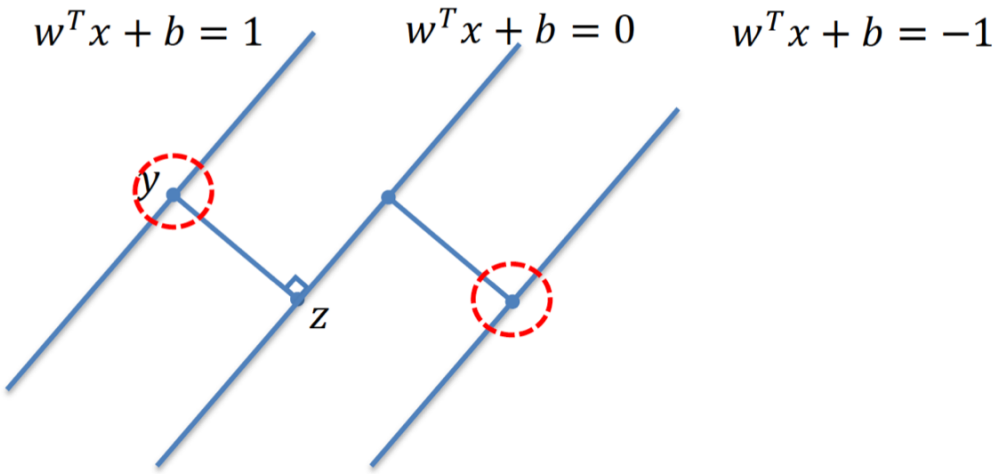
such that

$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \text{for all } i$$

- This is a quadratic optimization problem.
- Can be solved using packages of Python, Matlab, etc.

Classifier:

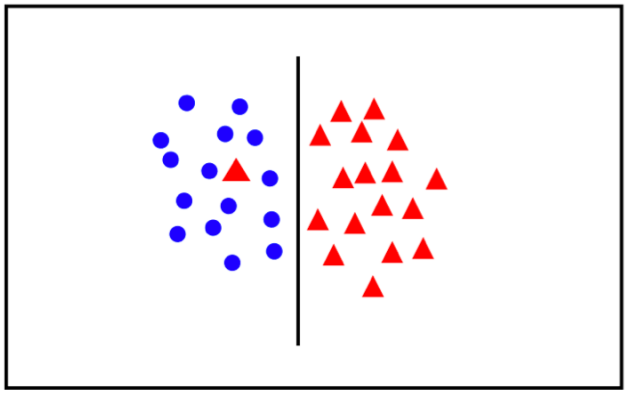
$$f(x) = w^T x + b$$

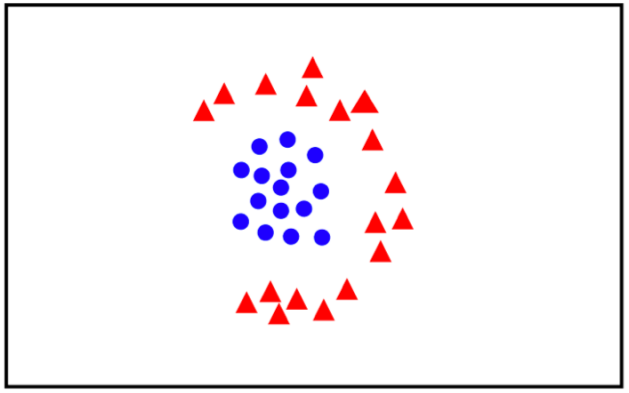


Where does the name come from?

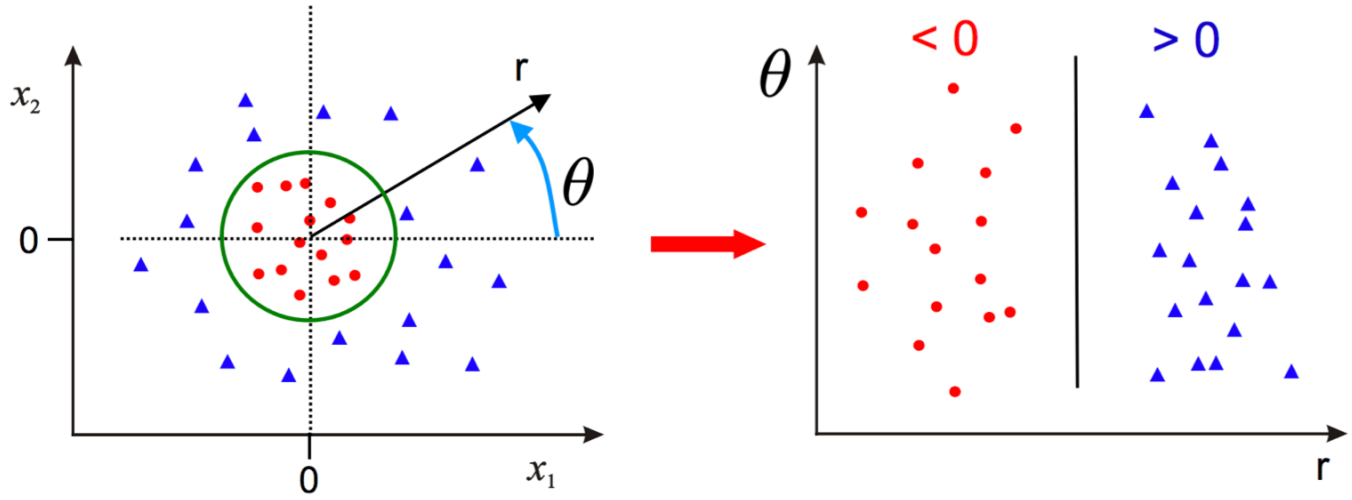
– The set of all data points such that $y^{(i)}(w^T x^{(i)} + b) = 1$ are called support vectors

What if data is not linearly separable?





What if data is not linearly separable?



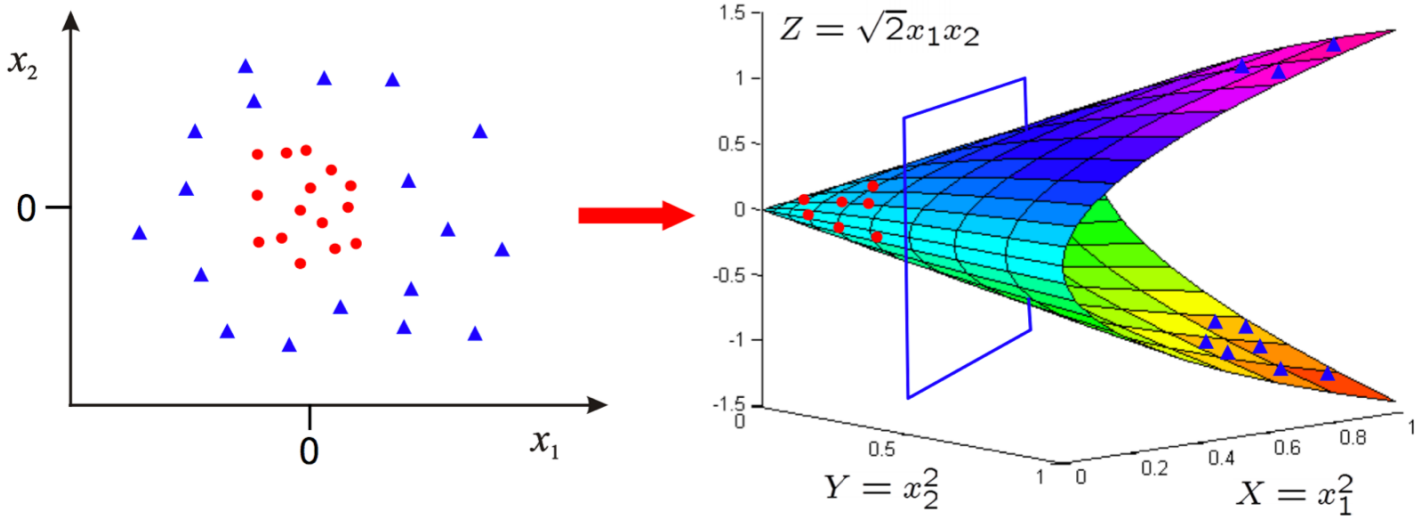
- Data is linearly separable in polar coordinates
- Acts non-linearly in original space

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} r \\ \theta \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

Feature map!

What if data is not linearly separable?

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



Disadvantage: High dimensions for complex problems

- Overfitting!
- Computationally expensive!

Kernel SVM

- Solve the optimization problem while avoiding the expensive dimensional transformation

- A mathematical detour
  - Lagrange multiplier
  - Dual problem

Lagrange multiplier

$$\min_{x \in \mathbb{R}^n} f_0(x)$$

subject to:

$$f_i(x) \leq 0, \qquad i = 1, \ldots, m$$
$$h_i(x) = 0, \qquad i = 1, \ldots, p$$

Lagrange multiplier

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

Incorporate constraints into a new objective function

$\lambda \geq 0$ and ν are vectors of ***Lagrange multipliers***

Lagrange multiplier

$$\min_{x \in \mathbb{R}^n} f_0(x)$$

subject to:

$$f_i(x) \leq 0, \qquad i = 1, \ldots, m$$
$$h_i(x) = 0, \qquad i = 1, \ldots, p$$

Equivalently,

$$\inf_x \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

Primal problem

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

Dual problem

$$\sup_{\lambda \geq 0, \nu} \inf_{x} L(x, \lambda, \nu)$$

Dual problem

$$\inf_{x} \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

Primal problem

$$\sup_{\lambda \geq 0, \nu} \inf_{x} L(x, \lambda, \nu) \leq \inf_{x} \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

Duality

Under certain conditions, the two optimization problems are equivalent

$$\sup_{\lambda \geq 0, \nu} \inf_{x} L(x, \lambda, \nu) = \inf_{x} \sup_{\lambda \geq 0, \nu} L(x, \lambda, \nu)$$

- This is called strong duality

Duality

- Slater's condition

$$\min_{x \in \mathbb{R}^n} f_0(x)$$

subject to:

$$f_i(x) \leq 0, \qquad i = 1, \ldots, m$$
$$Ax = b$$

where $f_0, \ldots, f_m$ are convex functions, strong duality holds if there exists an x such that

$$f_i(x) < 0, \qquad i = 1, \ldots, m$$
$$Ax = b$$

Back to SVM

$$\min_{w} \|w\|^2$$

such that

$$y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \text{for all } i$$

Slater's condition satisfied if x is feasible

Back to SVM

$$L(w, b, \lambda) = \frac{1}{2} w^T w + \sum_i \lambda_i (1 - y_i (w^T x^{(i)} + b))$$

Convex in w, so take derivatives to form the dual

$$\frac{\partial L}{\partial w_k} = w_k + \sum_i -\lambda_i y_i x_k^{(i)} = 0$$

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

Dual SVM

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j x^{(i)^T} x^{(j)} + \sum_i \lambda_i$$

such that

$$\sum_i \lambda_i y_i = 0$$

- By strong duality, solving this problem is equivalent to solving the primal problem
  - Given the optimal λ, we can easily construct w (b can be found by complementary slackness)

Dual SVM with feature map

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \Phi\left(x^{(i)}\right)^T \Phi\left(x^{(j)}\right) + \sum_i \lambda_i$$

such that

$$\sum_i \lambda_i y_i = 0$$

- The dual formulation only depends on inner products between the data points
  - Same thing is true if we use feature vectors instead

Kernel SVM

Feature map

Kernel function

Inner product

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \Phi\left(x^{(i)}\right)^T \Phi\left(x^{(j)}\right) + \sum_i \lambda_i$$

$$K\left(x^{(i)}, x^{(j)}\right)$$

Kernel trick:
compute inner product with kernel functions even if feature vectors are very large!

Mercer Condition

- Is there a mapping Φ(*x*) for any symmetric function *K*(x,*z*)? No
- The SVM dual formulation requires calculation *K(xi , xj)* for each pair of training instances. The array *Gij = K(xi , xj)* is called the Gram matrix
- There is a feature space Φ(*x*) when the Kernel is such that *G* is always semi-positive definite

Example: Polynomial Kernel

$$K(x, z) = (x * z + c)^p$$

is called the polynomial kernel of degree *p*.

- For *p*=2, feature map is

$$\varphi(x) = \langle x_n^2, \ldots, x_1^2, \sqrt{2}x_n x_{n-1}, \ldots, \sqrt{2}x_n x_1, \sqrt{2}x_{n-1}x_{n-2}, \ldots, \sqrt{2}x_{n-1}x_1, \ldots, \sqrt{2}x_2 x_1, \sqrt{2c}x_n, \ldots, \sqrt{2c}x_1, c\rangle$$

- If we measure 7,000 genes using the kernel once means calculating a summation product with 7,000 terms then taking the square of this number
- Mapping explicitly to the high-dimensional space means calculating approximately 50,000,000 new features for both training instances, then taking the inner product of that (another 50,000,000 terms to sum)
- In general, using the Kernel trick provides huge computational savings over explicit mapping!

Example: Gaussian Kernel

- **Consider the Gaussian kernel**

$$\exp\left(\frac{-\|x-z\|^2}{2\sigma^2}\right) = \exp\left(\frac{-(x-z)^T(x-z)}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-\|x\|^2 + 2x^Tz - \|z\|^2}{2\sigma^2}\right)$$

$$= \exp(-\|x\|^2)\exp(-\|z\|^2)\exp\left(\frac{x^Tz}{\sigma^2}\right)$$

- **Use the Taylor expansion for** $\exp()$

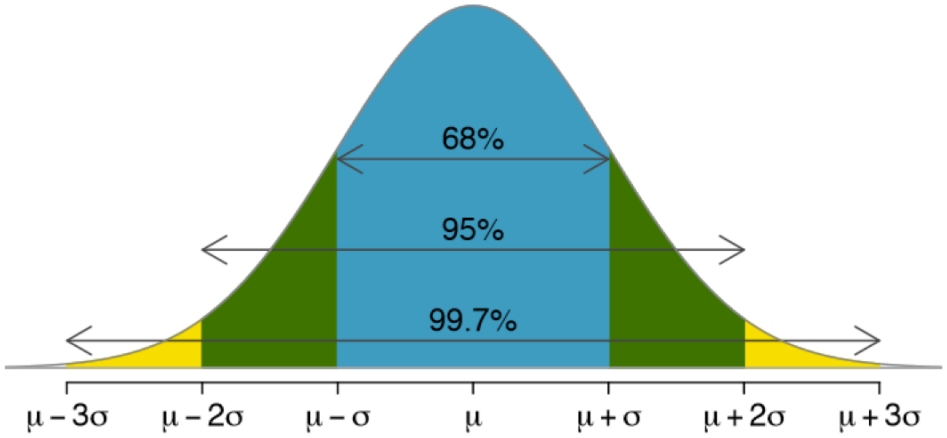
$$\exp\left(\frac{x^Tz}{\sigma^2}\right) = \sum_{n=0}^{\infty}\frac{(x^Tz)^n}{\sigma^{2n}n!}$$

Polynomial kernels of every degree!

Example: Gaussian Kernel

Kernel function: $\exp\left(\frac{-\|x-z\|^2}{2\sigma^2}\right)$ Similarity of data points

Objective function: $-\frac{1}{2}\sum_i\sum_j \lambda_i\lambda_j y_i y_j \Phi(x^{(i)})^T \Phi(x^{(j)}) + \sum_i \lambda_i$



σ: "detection radius"

- Small: local details
- Large: global picture

Kernel SVM

Learning:

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \Phi\left(x^{(i)}\right)^T \Phi\left(x^{(j)}\right) + \sum_i \lambda_i$$

such that

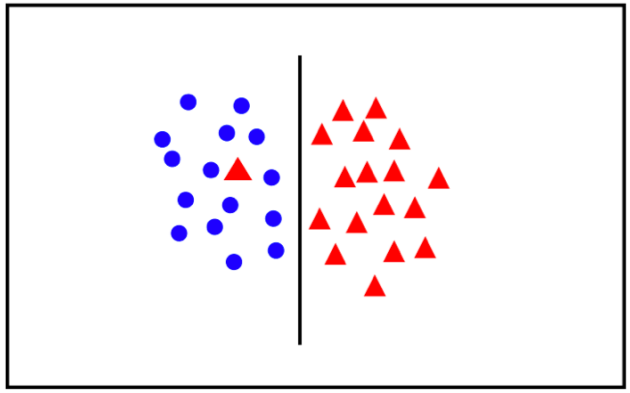
$$\sum_i \lambda_i y_i = 0$$

Classifier:

$$f(x) = \sum_i^M \lambda_i y_i \Phi\left(x^{(i)}\right)^T \Phi(x) + b$$

- Weighted votes
- Only support vectors matter? Convince yourself

SVM with slack



$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + c\sum_i \xi_i$$

such that

$$y_i\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, \text{for all } i$$

$$\xi_i \geq 0, \text{for all } i$$

Potentially allows some points to be misclassified/inside the margin

- How does this objective change with c?
  - As $c \to \infty$, requires a perfect classifier
  - As $c \to 0$, allows arbitrary classifiers (i.e., ignores the data)

- What is the optimal value of ξ for fixed w and b?
  - If $y_i\left(w^T x^{(i)} + b\right) \geq 1$, then $\xi_i = 0$
  - If $y_i\left(w^T x^{(i)} + b\right) < 1$, then $\xi_i = 1 - y_i\left(w^T x^{(i)} + b\right)$

- Obtain a new objective by substituting in for ξ

$$\min_{w,b} \frac{1}{2}\|w\|^2 + c\sum_i \max\left\{0, 1 - y_i\left(w^T x^{(i)} + b\right)\right\}$$

Penalty to prevent overfitting

Hinge loss

Dual of slack formulation

$$L(w, b, \xi, \lambda, \mu) = \frac{1}{2} w^T w + c \sum_i \xi_i + \sum_i \lambda_i (1 - \xi_i - y_i (w^T x^{(i)} + b)) + \sum_i -\mu_i \xi_i$$

Convex in w, b, ξ, so take derivatives to form the dual

$$\frac{\partial L}{\partial w_k} = w_k + \sum_i -\lambda_i y_i x_k^{(i)} = 0$$

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

$$\frac{\partial L}{\partial \xi_k} = c - \lambda_k - \mu_k = 0$$

Dual of slack formulation

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j x^{(i)^T} x^{(j)} + \sum_i \lambda_i$$

such that

$$\sum_i \lambda_i y_i = 0$$

$$c \geq \lambda_i \geq 0, \text{for all } i$$

Hyperparameters

- SVM: c and σ (if you choose Gaussian kernel)

- Separate data as training, validation and testing sets
  - Build model with training data
  - Choose hyperparameters with validation dataset
  - Evaluate performance with testing data

Suggested further reading

- http://www.kernel-machines.org/tutorial.html
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
- P.H. Chen, C.-J. Lin, and B. Schölkopf. A tutorial on nu -support vector machines. 2003.
- N. Cristianini. ICML'01 tutorial, 2001.
- K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181-201, May 2001. (PDF)
- B. Schölkopf. SVM and kernel methods, 2001. Tutorial given at the NIPS Conference.
- Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, Springel 2001