CSE 5526: Introduction to Neural Networks

Support Vector Machines (SVMs), Part 2

Back to SVMs: Maximum margin solution is a fixed point of the Lagrangian function

- Recall, the maximum margin hyperplane is

$$\text{argmin}_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}\|^2 \;\; \text{subject to } d_p\left(\boldsymbol{w}^T \boldsymbol{x}_p + b\right) \geq 1$$

  - Minimization of a quadratic function subject to multiple linear inequality constraints

- Will use Lagrange multipliers, a_p, to write Lagrangian function

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_p a_p (d_p\left(\boldsymbol{w}^T \boldsymbol{x}_p + b\right) - 1)$$

- Note that $\boldsymbol{x}_p$ and d_p are fixed for the optimization

Dual form of Lagrangian eliminates $\boldsymbol{w}$ and b

- Set derivatives of $L(\boldsymbol{w}, b, \boldsymbol{a})$ WRT $\boldsymbol{w}$ and b to 0

$$\frac{\partial}{\partial \boldsymbol{w}} L = 0 = \boldsymbol{w} - \sum_p a_p d_p \boldsymbol{x}_p$$

$$\Rightarrow \boldsymbol{w} = \sum_p a_p d_p \boldsymbol{x}_p$$

$$\frac{\partial}{\partial b} L = 0 = \sum_p a_p d_p$$

Dual form of Lagrangian eliminates $\boldsymbol{w}$ and b

- Note that:

$$\boldsymbol{w}^T\boldsymbol{w} = \sum_p a_p d_p \boldsymbol{w}^T \boldsymbol{x}_p = \sum_p \sum_q a_p a_q d_p d_q \boldsymbol{x}_p^T \boldsymbol{x}_q$$

- "Primal" form of Lagragian

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_p a_p(d_p\left(\boldsymbol{w}^T\boldsymbol{x}_p + b\right) - 1)$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_p a_p d_p w^T x_p - b\sum_p a_p d_p + \sum_p a_p$$

Dual form of Lagrangian eliminates $\boldsymbol{w}$ and b

$$L(\boldsymbol{w}, b, \boldsymbol{a})$$
$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_p a_p d_p w^T x_p - b\sum_p a_p d_p + \sum_p a_p$$
$$= \left(\frac{1}{2} - 1\right)\sum_p \sum_q a_p a_q\, d_p d_q \boldsymbol{x}_p^T \boldsymbol{x}_q - b \cdot 0 + \sum_p a_p$$

- So dual form of Lagrangian:

$$\tilde{L}(\boldsymbol{a}) = -\frac{1}{2}\sum_p \sum_q a_p a_q\, d_p d_q \boldsymbol{x}_p^T \boldsymbol{x}_q + \sum_p a_p$$

Dual form of Lagrangian eliminates $\boldsymbol{w}$ and b

- Dual form of Lagrangian, maximize:

$$\tilde{L}(\boldsymbol{a}) = -\frac{1}{2}\sum_p \sum_q a_p a_q\, d_p d_q \boldsymbol{x}_p^T \boldsymbol{x}_q + \sum_p a_p$$

- Subject to the constraints

$$a_p \geq 0\ \ \forall p \qquad \sum_p a_p d_p = 0$$

- Another quadratic programming problem subject to linear inequality and equality constraints

Dual form allows use of Kernel function

- In dual form, $\boldsymbol{x}_p$s only interact as inner products:

$$\tilde{L}(\boldsymbol{a}) = -\frac{1}{2}\sum_p \sum_q a_p a_q \, d_p d_q \boldsymbol{x}_p^T \boldsymbol{x}_q + \sum_p a_p$$

- Can replace $\boldsymbol{x}_p^T \boldsymbol{x}_q$ with kernel function $k\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right)$
- Think of kernel function as inner product of feature vector of $\boldsymbol{x}_p$s in some high dimensional space

$$k\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right) = \phi^T\left(\boldsymbol{x}_p\right)\phi\left(\boldsymbol{x}_q\right)$$

- But don't actually have to instantiate $\phi\left(\boldsymbol{x}_p\right)$
  - More about kernels shortly

Dual form is faster to solve when $D > N$

- Solving a quadratic program in M variables takes takes $O(M^3)$ time in general
- Primal form involves D variables ($\boldsymbol{w}$)
  - Dimensionality of the data $\boldsymbol{x}_p$,
  - Or dimensionality of features of the data $\phi(\boldsymbol{x}_p)$
- Dual form involves N variables ($\boldsymbol{a}$)
  - Number of training points
- SVMs are generally most useful with kernels
  - So $D > N$ and the dual is faster to solve

Classify new points using $y(\boldsymbol{x})$

- Actual prediction function is still
$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$
- Get $\boldsymbol{w}$ from primal Lagrangian
$$\boldsymbol{w} = \sum_p a_p d_p \boldsymbol{x}_p$$
- Will discuss b shortly, so
$$y(\boldsymbol{x}) = \sum_p a_p d_p \boldsymbol{x}_p^T \boldsymbol{x} + b$$

Classify new points using $y(\boldsymbol{x})$, with kernel

- With a kernel, $\boldsymbol{w}^T = \sum_p a_p d_p \phi(\boldsymbol{x}_p)$
- Actual prediction function is

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$$

$$= \sum_p a_p d_p \phi^T(\boldsymbol{x}_p) \phi(\boldsymbol{x}) + b$$

$$= \sum_p a_p d_p k(\boldsymbol{x}_p, \boldsymbol{x}) + b$$

- In practice, save all $\boldsymbol{x}_p$ with $a_p > 0$
  - And compute $k(\boldsymbol{x}_p, \boldsymbol{x})$ at test time

KKT Conditions

- In the case of SVMs, the KKT conditions are

$$a_p \geq 0$$

$$d_p y(\boldsymbol{x}_p) - 1 \geq 0$$

$$a_p\left(d_p y(\boldsymbol{x}_p) - 1\right) = 0$$

- So either $a_p = 0$ or $d_p y(\boldsymbol{x}_p) - 1 = 0$
  - Constraint from each point is either ignored or active
- When $a_p = 0$, $\boldsymbol{w}$ is independent of that point
- When $d_p y(\boldsymbol{x}_p) = 1$, that point is on the margin
  - It is a support vector
  - Thus only the support vectors contribute to $\boldsymbol{w}$

Compute b from support vectors

- Get b from support vectors, which have margin 1
- In the linear case, for a support vector $\boldsymbol{x}_q^s$

$$y(\boldsymbol{x}_q^s) = d_p = \boldsymbol{w}^T \boldsymbol{x}_q^s + b$$

$$b = d_q^s - \sum_p a_p d_p \boldsymbol{x}_p^T \boldsymbol{x}_q^s$$

- When using a kernel

$$b = d_q^s - \sum_p a_p d_p k(\boldsymbol{x}_p, \boldsymbol{x}_q^s)$$

- For numerical stability, average over all SVs

Summary so far

- Finding the maximum margin hyperplane has been formulated as a constrained quadratic program
  - Convex problem, well studied, easy conceptually to solve
- Can be solved in the primal or dual formulation
  - Dual formulation permits the use of kernel functions
- Only some data points contribute to the solution
  - The support vectors
- So far, only applies to linearly separable data