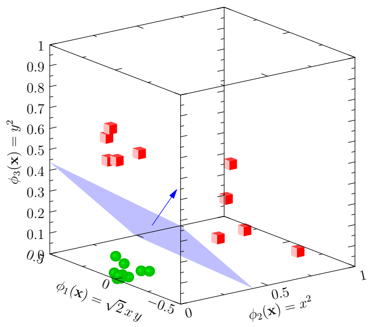
Advanced Machine Learning

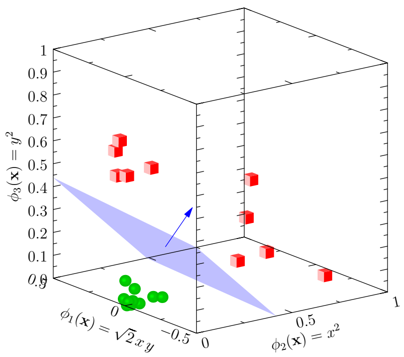
Dual Algorithm



Constrained Optimisation, Lagrangians, Dual Algorithm

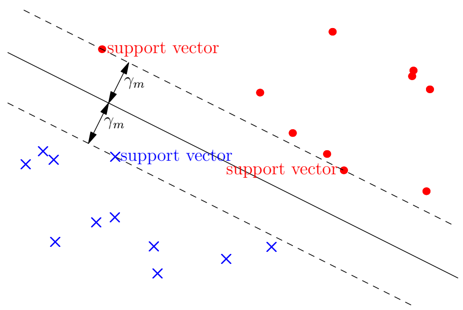
Outline

1. **Recap**
2. Constrained Optimisation
3. Duality



Linear SVM

- SVMs finds hyperplane which separates two sets of linearly separable data with maximum margin



Finding Maximum Margin Hyperplane

- We showed to find the maximum margin hyper-plane we can solve

$$\min_{\boldsymbol{w},b} \frac{\|\boldsymbol{w}\|^2}{2} \quad \text{subject to } y_k \left(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_k - b\right) \geq 1 \text{ for all } k = 1, 2, \ldots, m$$

(I've got rid of the hats on $\boldsymbol{w}$ and b because life is too short)

- This is a quadratic programme (a quadratic function with linear constraints)
- Quadratic programmes have a unique solution
- This is generally true of convex function optimisation

Extended Feature Spaces

- If we map into an extended feature space

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_p) \to \vec{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_r(\boldsymbol{x}))$$

$r \gg p$

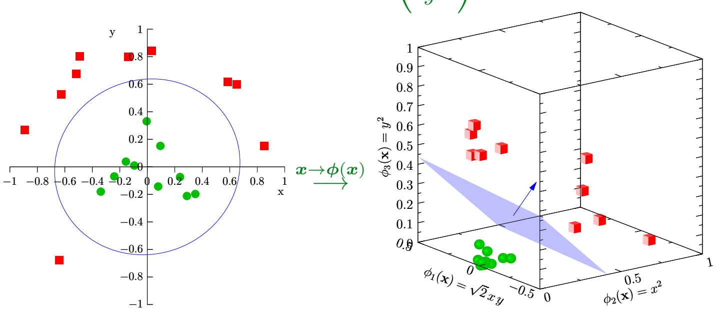
- The optimisation problem becomes

$$\min_{\vec{w},b} \frac{\|\vec{w}\|^2}{2} \quad \text{subject to } y_k \left(\vec{w}^\mathsf{T}\vec{\phi}(\boldsymbol{x}_k) - b\right) \geq 1 \text{ for all } k = 1, 2, \ldots, m$$

- $\vec{w}$ is an r dimensional vector (laying in the extended feature space)

Non-linearly Separation of Data

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix} \to \vec{\phi}(\boldsymbol{x}) \begin{pmatrix} 2\,x\,y \\ x^2 \\ y^2 \end{pmatrix}$$



Constrained Optimisation

- Recall that when we try to solve an optimisation problem with constraints we can add Lagrange multipliers
- The optimisation problem becomes

$$\min_{\vec{w},b} \frac{\|\vec{w}\|^2}{2} \quad \text{subject to } y_k \left(\vec{w}^\mathsf{T}\vec{\phi}(\boldsymbol{x}_k) - b\right) \geq 1 \text{ for all } k = 1, 2, \ldots, m$$

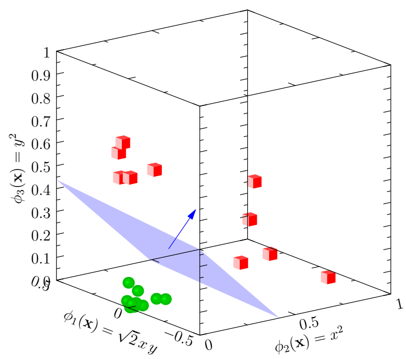
- Is equivalent to finding the extremal point of the Lagrangian

$$\mathcal{L}(\vec{w}, b, \alpha) = \frac{\|\vec{w}\|^2}{2} + \sum_{k=1}^{m} \alpha_k \left(y_k \left(\vec{w}^\mathsf{T}\vec{\phi}(\boldsymbol{x}_k) - b\right) - 1 \right)$$

- subject to $\alpha_k \geq 0$

Outline

1. Recap
2. **Constrained Optimisation**
3. Duality



Solving Constrained Optimisation Problems

- Suppose we have a problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to } g(\boldsymbol{x}) = 0$$

- A standard procedure is to define the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha\, g(\boldsymbol{x})$$

  where α is known as a Lagrange multiplier

- In the extended space $(\boldsymbol{x}, \alpha)$ we have to solve

$$\max_{\alpha} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \alpha)$$

Conditions on Optimum

- The optimisation problem is

$$\max_{\alpha} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \alpha) \quad \text{where} \quad \mathcal{L}(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha\, g(\boldsymbol{x})$$

- Assuming differentiability

$$\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \alpha) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) - \alpha\, \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) = 0$$
$$\frac{\partial \mathcal{L}}{\partial \alpha} = g(\boldsymbol{x}) = 0$$

- The second condition is just the constraint

Note on Gradients

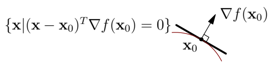
- Note that for any function $f(\boldsymbol{x})$ we can Taylor expand around $\boldsymbol{x}_0$

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathsf{T}} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathsf{T}} \mathbf{H} (\boldsymbol{x} - \boldsymbol{x}_0) + \ldots$$

  where $\mathbf{H}$ is a matrix of second derivative known as the Hessian

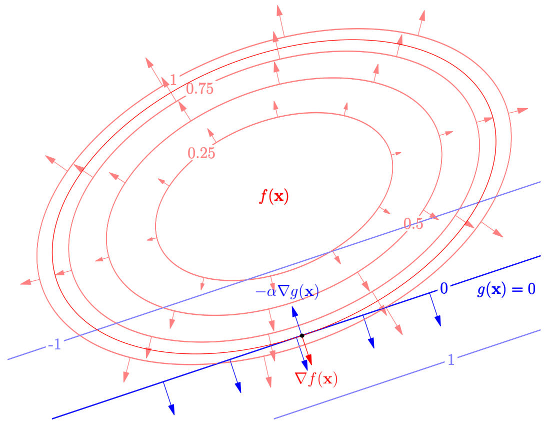
- If we consider points perpendicular to $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0)$ which go through $\boldsymbol{x}_0$ these will have values

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + O(\|\boldsymbol{x} - \boldsymbol{x}_0\|^2)$$



  thus $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ is always orthogonal to the contour lines

Constrained Optima



Example

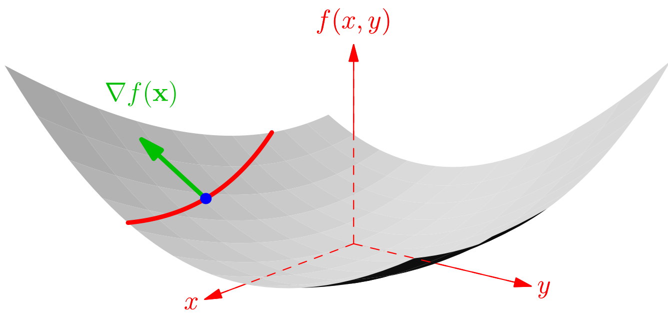
- Minimise $f(\boldsymbol{x}) = x^2 + 2\,y^2 - x\,y$
- Subject to $g(\boldsymbol{x}) = x - 2\,y - 3 = 0$
- Writing $\mathcal{L} = f(\boldsymbol{x}) - \alpha\, g(\boldsymbol{x})$
- Condition for minima is $\nabla_{\boldsymbol{x}} \mathcal{L} = 0$

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \begin{pmatrix} 2\,x - y \\ -x + 4\,y \end{pmatrix} = \alpha \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) = \alpha \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

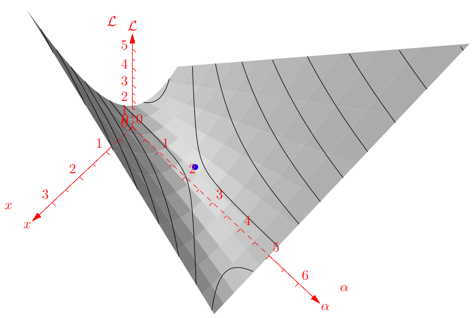
  and $\frac{\partial \mathcal{L}}{\partial \alpha} = g(\boldsymbol{x}) = x - 2\,y - 3 = 0$

- Solving simultaneous equations gives minima at $(x, y) = (\frac{3}{4}, -\frac{9}{8})$ with $\alpha = \frac{21}{8}$

Surface



Saddle-Point $y = -9/8$



Multiple Constraints

- Given an optimisation problem with multiple constraints

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to } g_k(\boldsymbol{x}) = 0 \text{ for } k = 1, 2, \ldots, m$$

- We introduce multiple Lagrange multipliers

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) - \sum_{k=1}^{m} \alpha_k\, g_k(\boldsymbol{x})$$

- The condition for an optima is $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}) = 0$ which implies

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{k=1}^{m} \alpha_k\, \nabla_{\boldsymbol{x}} g_k(\boldsymbol{x})$$

  plus the original constraints $\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha})}{\partial \alpha_k} = g_k(\boldsymbol{x}) = 0$

Example

- Minimise $f(\boldsymbol{x}) = x^2 + 2\,y^2 + 5\,z^2 - x\,y - x\,z$ subject to $g_1(\boldsymbol{x}) = x - 2\,y - z - 3 = 0$ and $g_2(\boldsymbol{x}) = 2\,x + 3\,y + z - 2 = 0$
- Writing $\mathcal{L}(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha_1\, g_1(\boldsymbol{x}) - \alpha_2\, g_2(\boldsymbol{x})$
- Condition for minima is $\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L} = 0$ or $\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{k=1}^{2} \alpha_k \boldsymbol{\nabla}_{\boldsymbol{x}} g_k(\boldsymbol{x})$

$$\begin{pmatrix} 2\,x - y - z \\ -x + 4\,y \\ 10\,z - x \end{pmatrix} = \alpha_1 \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

and $\frac{\partial \mathcal{L}}{\partial \alpha_i} = g_i(\boldsymbol{x}) = 0$

- Solving simultaneous equations gives minima at $(\frac{37}{20}, -\frac{11}{20}, -\frac{1}{20})$ with $\alpha_1 = 3$ and $\alpha_2 = \frac{13}{20}$

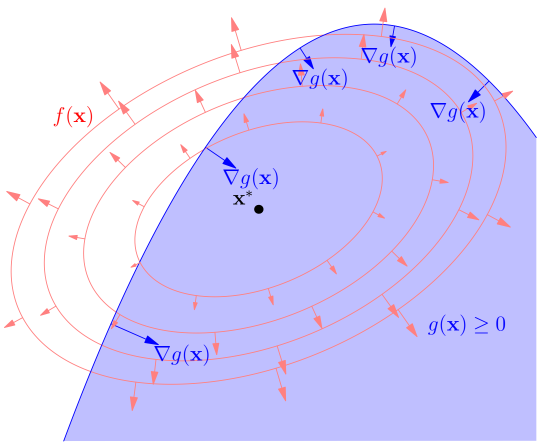
Inequality Constraints

- Suppose we have the problem

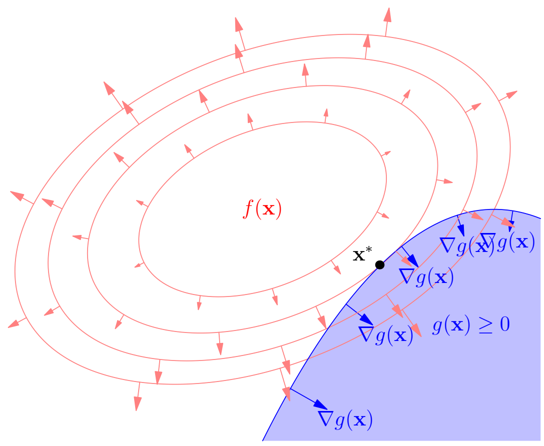
$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to } g(\boldsymbol{x}) \geq 0$$

- Looks much more complicated, but
- Only two things can happen
  - Either the minimum of $f(\boldsymbol{x})$, $\boldsymbol{x}^*$, satisfies $g(\boldsymbol{x}^*) > 0$
    - We then have an unconstrained optimisation problem
  - Otherwise, it satisfies $g(\boldsymbol{x}^*) = 0$
    - We have a constrained optimisation problem

Inside Region



On the Boundary



KKT Conditions

- To minimise $f(\boldsymbol{x})$ subject to $g(\boldsymbol{x}) \geq 0$

$$\mathcal{L}(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha\, g(\boldsymbol{x})$$

- Then $\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L} = 0$ or

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L} = \boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) - \alpha\, \boldsymbol{\nabla}_{\boldsymbol{x}} g(\boldsymbol{x}) = 0$$

- where either
  - $\alpha = 0$ and the solutions in the interior or
  - $\alpha > 0$ and $g(\boldsymbol{x}) = 0$, i.e. the solution is on the boundary
- These conditions are known as the Karush-Kuhn-Tucker conditions

Many Inequalities

- Given the problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to } g_k(\boldsymbol{x}) \geq 0 \text{ for } k = 1, 2, \ldots, m$$

- We introduce multiple Lagrange multipliers

$$\mathcal{L}(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \sum_{k=1}^{m} \alpha_k\, g_k(\boldsymbol{x})$$

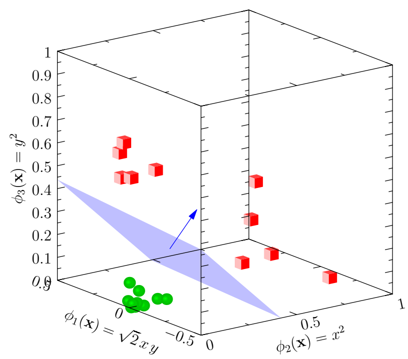
- The condition for an optima is

$$\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{k=1}^{m} \alpha_k\, \boldsymbol{\nabla}_{\boldsymbol{x}} g_k(\boldsymbol{x})$$

- Plus the constraints that either $\alpha_k = 0$ or $\alpha_k > 0$ and $g_k(\boldsymbol{x}) = 0$

Outline

1. Recap
2. Constrained Optimisation
3. **Duality**



Back to the SVM

- We showed that the quadratic programming problem can be written as

$$\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{w}, b} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) \quad \text{subject to } \alpha_k \geq 0$$

- Where the Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{k=1}^{m} \alpha_k \left(y_k \left(\boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_k - b \right) - 1 \right)$$

- $\boldsymbol{\nabla}_{\boldsymbol{w}} \mathcal{L} = \boldsymbol{w} - \sum_{k=1}^{m} \alpha_k\, y_k\, \boldsymbol{x}_k = 0$ implies that $\boldsymbol{w}^* = \sum_{k=1}^{m} \alpha_k\, y_k\, \boldsymbol{x}_k$
- $\frac{\partial \mathcal{L}}{\partial b} = \sum_{k=1}^{m} \alpha_k\, y_k = 0$ implies $\sum_{k=1}^{m} \alpha_k\, y_k = 0$

Dual Form of Quadratic Programming

- Substituting in $\boldsymbol{w}^*$ the optimisation condition becomes

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \, \alpha_l \, y_k \, y_l \, \boldsymbol{x}_k^{\mathsf{T}} \boldsymbol{x}_l$$

- Subject to $\sum_{k=1}^{m} \alpha_k \, y_k = 0$ and $\alpha_k \geq 0$ for all k
- This is also a quadratic programming problem!
- It is known as the *dual* form and depends on the number of training examples (it is solved by a standard package)
- It doesn't involve $\boldsymbol{w}$ and involves only the dot products $\boldsymbol{x}_k^{\mathsf{T}} \boldsymbol{x}_l$

Duality

- We can either work in the n-dimensional weight space

$$\min_{\boldsymbol{w},\, b} \frac{\|\boldsymbol{w}\|^2}{2} \quad \text{subject to } y_k \left(\boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_k - b \right) \geq 1 \text{ for all } k = 1, 2, \ldots, m$$

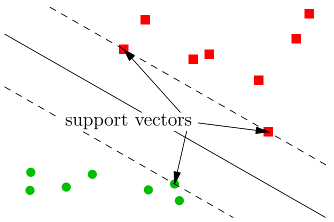
- Or the m-dimensional Lagrange multiplier space

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \, \alpha_l \, y_k \, y_l \, \boldsymbol{x}_k^{\mathsf{T}} \boldsymbol{x}_l \quad \text{with} \sum_{k=1}^{m} \alpha_k \, y_k = 0 \;\&\; \alpha_k \geq 0$$

- Both require sophisticated quadratic programming algorithms to solve
- Which is easiest depends or whether m is greater or less than p

Support Vectors

- Where $\alpha_k > 0$ the constraints are exactly met so $y_k(\boldsymbol{x}_k^{\mathsf{T}} \boldsymbol{w} - b) = 1$
- These data points are known as *support vectors*

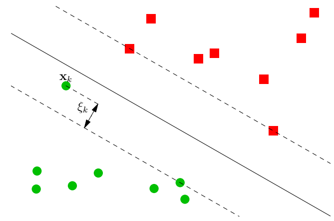


- In high dimensions there can be many support vectors

Soft Margins

- Recall we can relax constraints by introducing *slack variables*, $\xi_k \geq 0$

$$y_k(\boldsymbol{x}_k^{\mathsf{T}} \boldsymbol{w} - b) \geq 1 - \xi_k$$



- Minimise $\frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{k=1}^{m} \xi_k$
- subject to $\xi_k \geq 0$

Dealing with Slack Variables

- Our new Lagrangian is

$$\mathcal{L} = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{k=1}^{n} \xi_k - \sum_{k=1}^{m} \alpha_k \left(y_k \left(\boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_k - b \right) - 1 + \xi_k \right) - \sum_{k=1}^{m} \beta_k \, \xi_k$$

- Minimise with respect to ξ_k

$$\frac{\partial \mathcal{L}}{\partial \xi_k} = C - \alpha_k - \beta_k = 0$$

- But $\beta_k \geq 0 \quad \Rightarrow \quad \alpha_k \leq C$
- Thus $0 \leq \alpha_k \leq C$

Extended Feature Space

- In the extended feature space

$$\min_{\vec{w},\, b} \frac{\|\vec{w}\|^2}{2} \quad \text{subject to } y_k \left(\vec{w}^{\mathsf{T}} \vec{\phi}(\boldsymbol{x}_k) - b \right) \geq 1 \text{ for all } k = 1, 2, \ldots, m$$

- Giving the dual problems

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \, \alpha_l \, y_k \, y_l \, \vec{\phi}(\boldsymbol{x}_k)^{\mathsf{T}} \, \vec{\phi}(\boldsymbol{x}_l) \quad \text{with} \sum_{k=1}^{m} \alpha_k \, y_k = 0 \;\&\; \alpha_k \geq 0$$

- When $\vec{\phi}(\boldsymbol{x}_k)^{\mathsf{T}} \, \vec{\phi}(\boldsymbol{x}_l) = K(\boldsymbol{x}_k,\, \boldsymbol{x}_l)$ then

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \, \alpha_l \, y_k \, y_l \, K(\boldsymbol{x}_k,\, \boldsymbol{x}_l) \quad \text{with} \sum_{k=1}^{m} \alpha_k \, y_k = 0 \;\&\; \alpha_k \geq 0$$

Classifying New Data

- Having trained the SVM we now have to use it
- Given a new input $\boldsymbol{x}$ we decide on the class

$$\operatorname{sgn}\left(\vec{w}^{\mathsf{T}} \vec{\phi}(\boldsymbol{x}) - b \right) \qquad \text{but} \qquad \vec{w} = \sum_{k=1}^{m} \alpha_k \, y_k \, \vec{\phi}(\boldsymbol{x}_k)$$

- In the dual representation this becomes

$$\operatorname{sgn}\left(\sum_{k=1}^{m} \alpha_k \, y_k \, K(\boldsymbol{x}_k, \boldsymbol{x}) - b \right)$$

where we only need to sum over the non-zero α_k (i.e. the support vectors SVs)

Conclusion

- We can solve for the maximum-margin hyper-plane either in the primal form (space of weights and bias) or the dual form (space of Lagrange multipliers)
- In the dual form the solution only depends on the dot product $\boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j$ or $\vec{\phi}(\boldsymbol{x}_i)^{\mathsf{T}} \vec{\phi}(\boldsymbol{x}_j)$
- If $K(\boldsymbol{x},\, \boldsymbol{y}) = \vec{\phi}(\boldsymbol{x})^{\mathsf{T}} \, \vec{\phi}(\boldsymbol{y})$ we never have to explicitly compute $\vec{\phi}(\boldsymbol{x})$—the kernel trick
- This allows us to work in a very high dimensional space (giving low bias) while finding a maximum-margin hyper-plane (simple machine with low variance) magic!