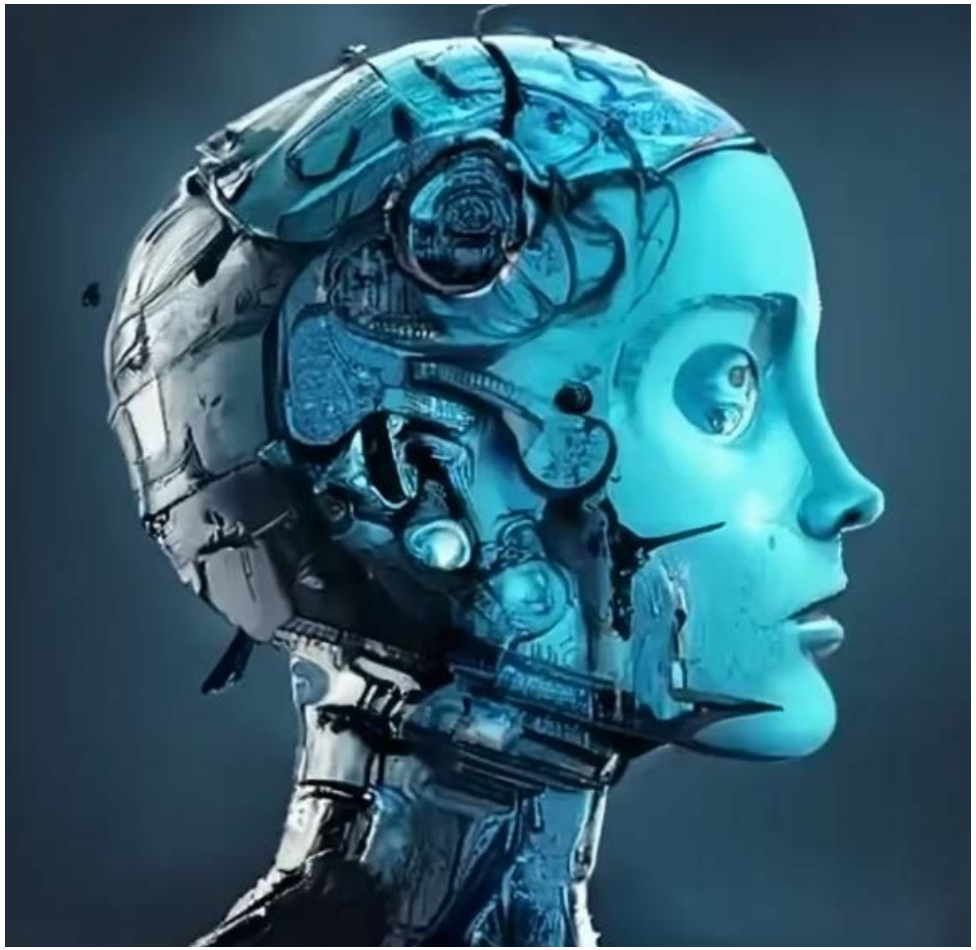
Machine Intelligence

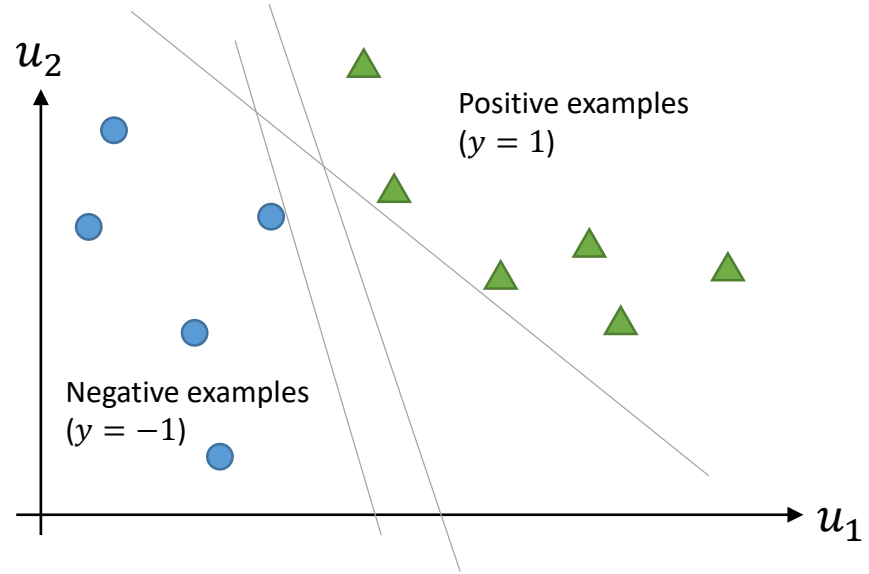
Support Vector Machines

Dr. Cory Merkel



Support Vector Machine Motivation

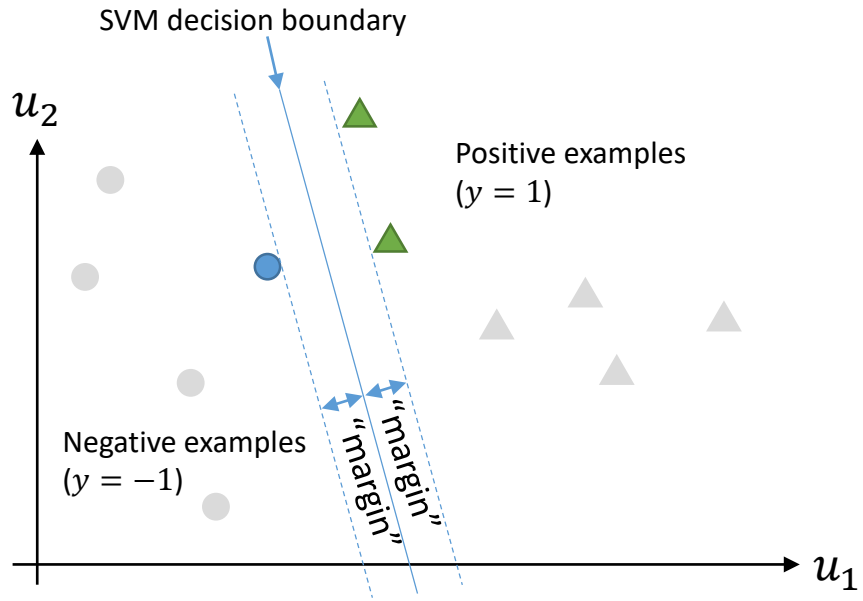
- Consider the following binary classification problem
- Which decision boundary is best?
  - There are infinite boundaries that will give 100% training accuracy, but which is likely to give the best test accuracy?
- Previously, we tried to maximize the *average* distance between the decision boundary and each datapoint (logistic regression)



- What if we, instead maximize the *minimum* distance between the decision boundary and the datapoints?

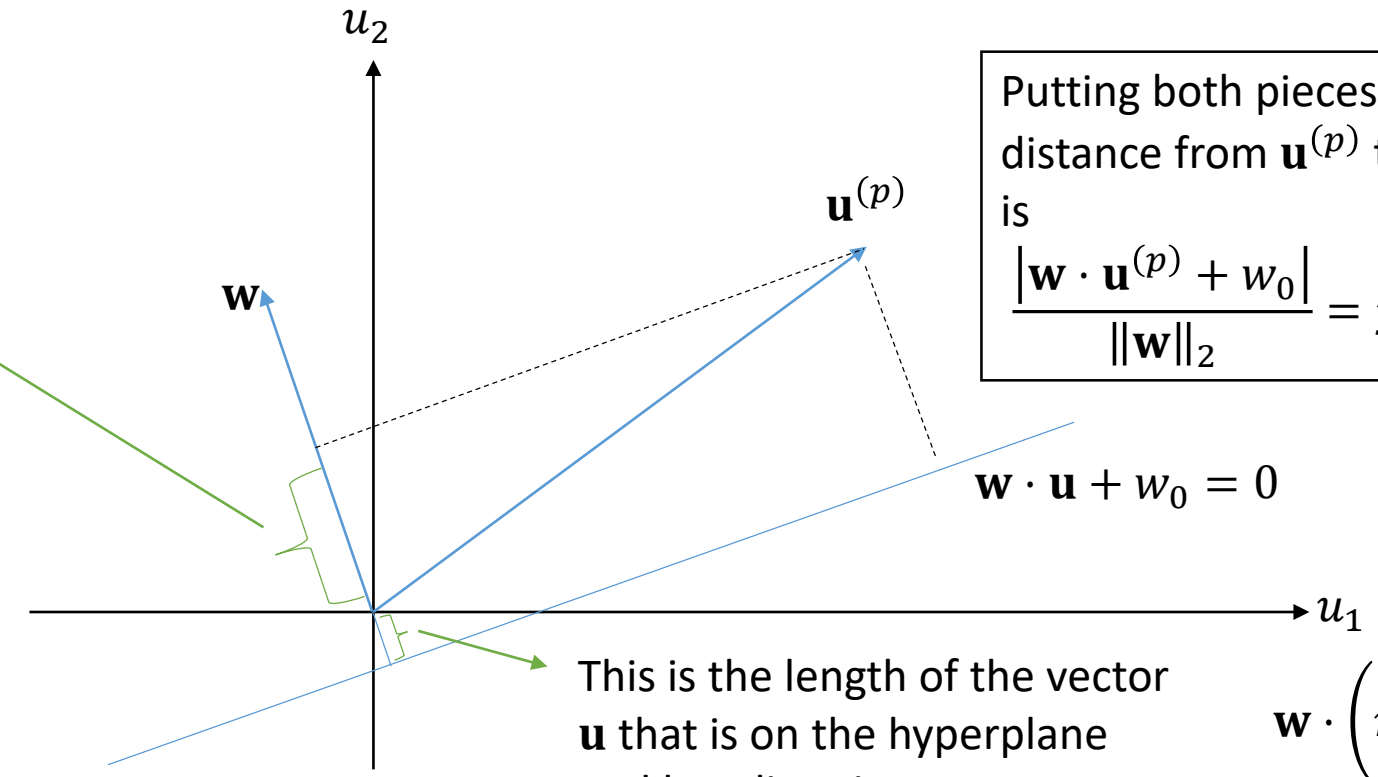
Support Vector Machine

- While the average distance depends on all datapoints, the minimum distance only depends on a few:
  - We call the set of the closest positive and negative examples the *support vectors*
  - The distance between the decision boundary and support vectors is called the margin
- We want to maximize the margin
  - Need to have an expression for the distance from the decision boundary to the datapoints



Support Vector Machine

- What is the Euclidean distance between the hyperplane $\hat{y} = \mathbf{w} \cdot \mathbf{u} + w_0 = 0$ and a point $\mathbf{u}^{(p)}$?



This is just the length of the projection of $\mathbf{u}^{(p)}$ onto $\mathbf{w}$:

$$\left\|\text{proj}_{\mathbf{w}}\mathbf{u}^{(p)}\right\|_2 = \frac{\left|\mathbf{w} \cdot \mathbf{u}^{(p)}\right|}{\|\mathbf{w}\|_2}$$

Putting both pieces together, the distance from $\mathbf{u}^{(p)}$ to the hyperplane is

$$\frac{\left|\mathbf{w} \cdot \mathbf{u}^{(p)} + w_0\right|}{\|\mathbf{w}\|_2} = y^{(p)} \frac{\mathbf{w} \cdot \mathbf{u}^{(p)} + w_0}{\|\mathbf{w}\|_2}$$

This is the length of the vector $\mathbf{u}$ that is on the hyperplane and has direction $\mathbf{w}$:

$$\mathbf{w} \cdot \left(r \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\right) + w_0 = 0$$

$$|r| = |w_0| / \|\mathbf{w}\|_2$$

Support Vector Machine

- We can now formulate the optimization problem as

$$\arg\max_{\mathbf{w},w_0} \left\{ \min_p y^{(p)} \frac{\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0}{\|\mathbf{w}\|_2} \right\}$$

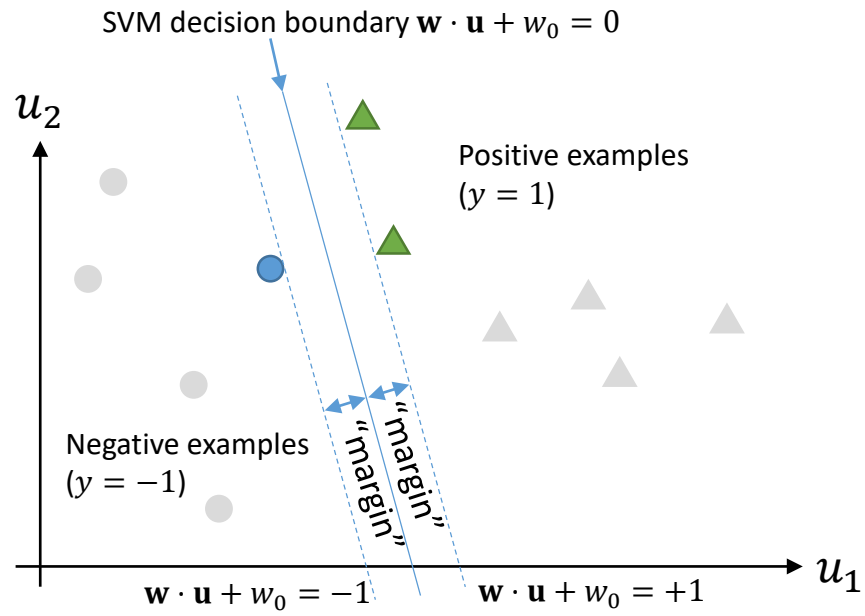
$$= \arg\max_{\mathbf{w},w_0} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_p y^{(p)} \left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \right\}$$

- For simplicity, let's also say that we want our $\mathbf{w}$ and w_0 to be scaled such that

$$\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0 \geq 1 \quad \text{when } y^{(p)} = 1$$
$$\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0 \leq -1 \quad \text{when } y^{(p)} = -1$$

Or, equivalently:

$$y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \geq 1$$



Support Vector Machine

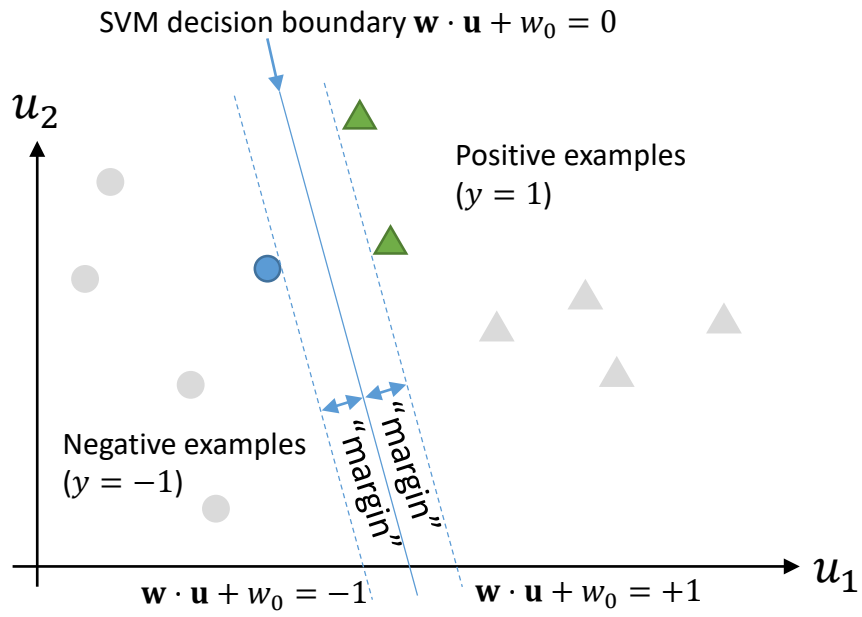
- Our optimization problem can be rewritten from:

$$\arg\max_{\mathbf{w},w_0} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_p y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \right\}$$

to

$$\text{minimize:} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$

$$\text{subject to:} \quad y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \geq 1 \;\forall\, p$$



- Note that we have converted our arg max into a minimization and we have converted our min into a set of constraints!

Forming the Lagrangian

- Often, when we seek an optimum of a function $f(\mathbf{x})$ subject to constraints $g_j(\mathbf{x}) \leq 0$, $h_k(\mathbf{x}) = 0$, we can find the solution by optimizing the linear combination of the function and the constraints:

$$L = f(\mathbf{x}) + \sum_j \alpha_j g_j(\mathbf{x}) + \sum_k \beta_k h_k(\mathbf{x})$$

where α_j, β_k are constants called Lagrange multipliers

  - L is called the *Lagrangian*

- It is kind of like we are modifying a loss function with terms that tell us how close we are to meeting our constraints.

Forming the Lagrangian

- Our problem

$$\text{minimize:} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$

$$\text{subject to:} \quad y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \geq 1 \;\forall\; p$$

has Lagrangian: $L = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_p \alpha_p\left[1 - y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right)\right]$

- To find the optimum, we set the derivative equal to 0:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_p \alpha_p y^{(p)}\mathbf{u}^{(p)} = 0 \qquad \frac{\partial L}{\partial w_0} = \sum_p \alpha_p y^{(p)} = \boldsymbol{\alpha}\cdot\mathbf{y} = 0$$

- So, we can get $\mathbf{w}$ if we can find α_p

The Dual Problem

- Our original problem (minimize … subject to …) is often called the *primal* problem. It's the one we want to solve!
- We can use our Lagrangian formulation to come up with an equivalent *dual* problem
- Using our values for $\mathbf{w}$ and w_0 (previous slide), we can write the Lagrangian as

$$L = \sum_p \alpha_p - \sum_{p,q} \alpha_p y^{(p)} \left(\mathbf{u}^{(p)} \cdot \mathbf{u}^{(q)} \right) y^{(k)} \alpha_k$$

$$= \mathbf{1} \cdot \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha} \, \mathrm{diag}(\mathbf{y}) \, \mathbf{G} \, \mathrm{diag}(\mathbf{y}) \, \boldsymbol{\alpha}$$

Vector of 1s

Gram Matrix:
$\mathbf{G}_{p,q} \equiv \mathbf{u}^{(p)} \cdot \mathbf{u}^{(q)}$

The Dual Problem

- Now, our optimization problem becomes:

$$L = \mathbf{1} \cdot \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha} \operatorname{diag}(\mathbf{y}) \mathbf{G} \operatorname{diag}(\mathbf{y}) \boldsymbol{\alpha}$$

- Now, we apply a very important result called the Karush-Kuhn-Tucker conditions that says our dual problem can be formulated as

$$\text{minimize: } \frac{1}{2} \boldsymbol{\alpha} \operatorname{diag}(\mathbf{y}) \mathbf{G} \operatorname{diag}(\mathbf{y}) \boldsymbol{\alpha} - \mathbf{1} \cdot \boldsymbol{\alpha}$$

$$\text{subject to: } \begin{array}{l} y^{(p)}\left(\mathbf{w} \cdot \mathbf{u}^{(p)} + w_0\right) - 1 \geq 0 \\ \alpha_p \geq 0 \\ \alpha_p \left[y^{(p)}\left(\mathbf{w} \cdot \mathbf{u}^{(p)} + w_0\right) - 1\right] = 0 \\ \boldsymbol{\alpha} \cdot \mathbf{y} = 0 \end{array}$$

KKT Conditions

- So, we can get our Lagrange multipliers from solving this dual problem, and then get $\mathbf{w}$ and w_0 using

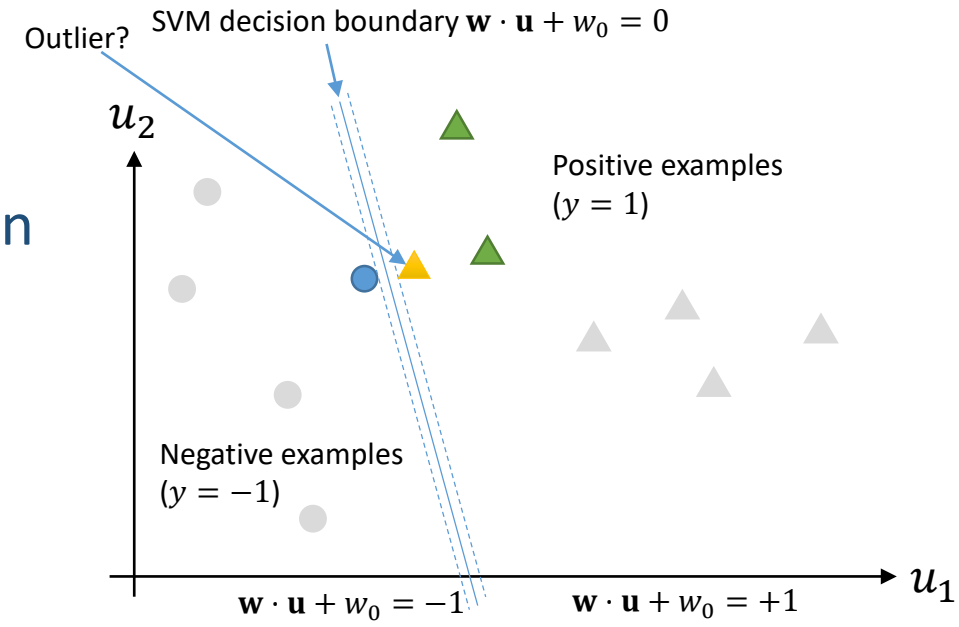
$$\mathbf{w} = \sum_p \alpha_p y^{(p)} \mathbf{u}^{(p)}$$

$$\alpha_p \left[y^{(p)}\left(\mathbf{w} \cdot \mathbf{u}^{(p)} + w_0\right) - 1\right] = 0$$

- Use a standard SVM package to solve this! (e.g. libsvm, scikit-learn, etc.)

Soft Margin SVM

- What happens if most of our data have a good (large) margin, but there appear to be some outliers?
  - If we use our previous algorithm, the decision boundary may have a very small margin
  - Let's allow our SVM to ignore these and let some datapoints lie inside the margin



Soft Margin SVM

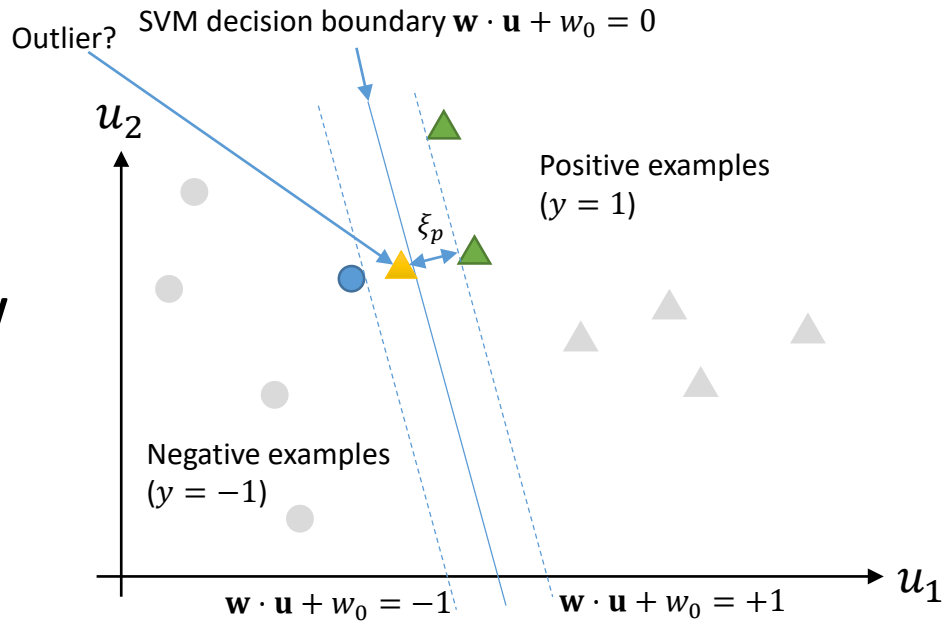
- Previously, we had:

$$y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)}+w_0\right)\geq 1$$

- Now, we relax this condition:

$$y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)}+w_0\right)\geq 1-\xi_p, \xi_p\geq 0$$

  where ξ_p is a *slack variable* that tells us how far $\mathbf{u}^{(p)}$ is allows to move into the margin.
  - Note, if $\xi_p < 1$, $\mathbf{u}^{(p)}$ will still be classified correctly



Soft Margin SVM

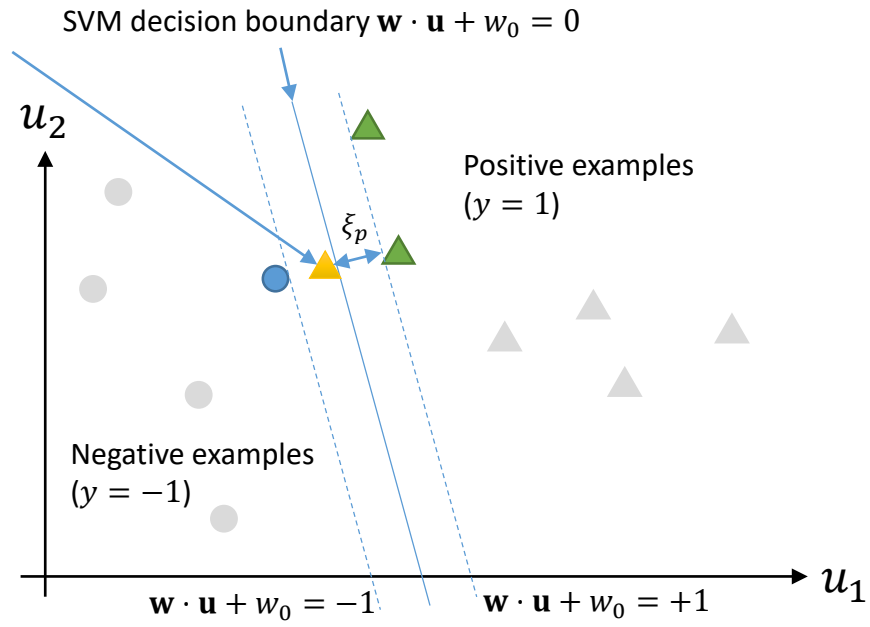
- Now, our optimization problem becomes

$$\text{minimize:} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_p \xi_p$$

$$\text{subject to:} \quad y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) \geq 1 - \xi_p \;\forall\, p$$

$$\xi_p \geq 0$$

- The parameter C controls how much slack we allow and acts as a regularizer



Soft Margin SVM

- Our new Lagrangian is

$$L = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_p \xi_p + \sum_p \alpha_p\left[1 - y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right)\right] - \sum_p \beta_p\xi_p$$

- Note that we now have two sets of Lagrange multipliers, α and β since we have two different constraints
- The KKT conditions are

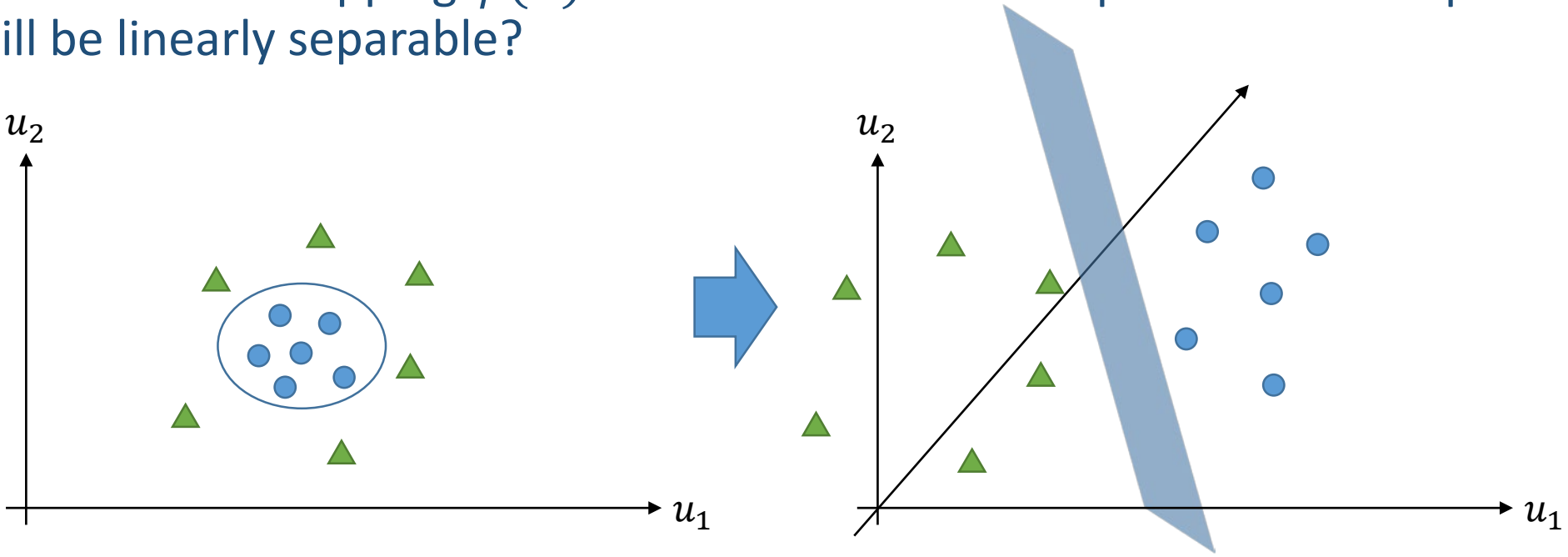
$$\begin{aligned}
\alpha_p, \beta_p, \xi_p &\geq 0 \\
y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) - 1 + \xi_p &\geq 0 \\
\alpha_p\left[y^{(p)}\left(\mathbf{w}\cdot\mathbf{u}^{(p)} + w_0\right) - 1 + \xi_p\right] &= 0 \\
\beta_p\xi_p &= 0
\end{aligned}$$

- Solve by 1.) finding stationarity conditions: $\frac{\partial L}{\partial \mathbf{w}}, \frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial \xi_p} = 0$, 2.) Formulate

  and solve the dual problem for Lagrange multipliers, 3.) use stationarity and KKT conditions to get $\mathbf{w}$ and w_0

Kernel Trick

- What if our data are not linearly separable?
  - For the following case, we'd like some kind of oval decision boundary, but we don't want to lose all the nice properties of the *linear* algebra we have been doing
  - Is there some mapping $\phi(\mathbf{u})$ into a new *embedded* space where the points will be linearly separable?



Kernel Trick

- Now, our optimization problem is

$$\text{minimize:} \quad \frac{1}{2}\|\tilde{\mathbf{w}}\|_2^2 + C\sum_p \tilde{\xi}_p$$

$$\text{subject to:} \quad y^{(p)}\left(\tilde{\mathbf{w}} \cdot \phi\left(\mathbf{u}^{(p)}\right) + \tilde{w}_0\right) \geq 1 - \tilde{\xi}_p \;\forall\; p$$

$$\tilde{\xi}_p \geq 0$$

- Note that the mapping $\phi(\mathbf{u})$ typically increases the dimensionality substantially: $\dim(\tilde{\mathbf{w}}) \gg \dim(\mathbf{w})$
  - **However, the dual form of the problem turns out to be much simpler!**

Kernel Trick – Dual Problem

- The dual form is:

minimize: $\frac{1}{2}\boldsymbol{\alpha} \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{K} \cdot \mathrm{diag}(\mathbf{y}) \cdot \boldsymbol{\alpha} + \mathbf{1} \cdot \boldsymbol{\alpha}$

subject to: $0 \leq \alpha_p \leq C$

$\boldsymbol{\alpha} \cdot \mathbf{y} = 0$

where $\mathbf{K}$ is there *kernel* matrix: $K_{pq} \equiv \phi\left(\mathbf{u}^{(p)}\right) \cdot \phi\left(\mathbf{u}^{(q)}\right)$

Common Kernels

- Linear: $K_{pq} = \mathbf{u}^{(p)} \cdot \mathbf{u}^{(q)}$
- Polynomial: $K_{pq} = \left(1 + \mathbf{u}^{(p)} \cdot \mathbf{u}^{(q)}\right)^{Q}$
- Sigmoid: $K_{pq} = \tanh\left(a\mathbf{u}^{(p)} \cdot \mathbf{u}^{(q)} - b\right)$
- Radial Basis Function: $K_{pq} = \exp\left(-\frac{\left(\mathbf{u}^{(p)} - \mathbf{u}^{(q)}\right)^2}{2\sigma^2}\right)$