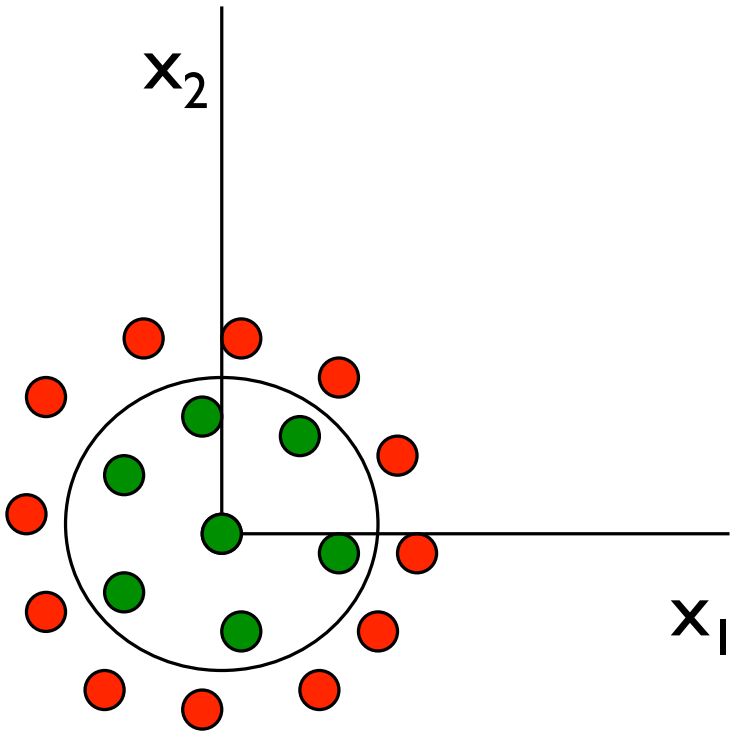
Recap of models for classification

Given $\mathcal{D} = \{(x^{(i)}, y^{(i)}) | 1 \leq i \leq m, x^{(i)} \in \Re^d, y^{(i)} \in \{0, 1\}\}$, we have identified two parametric classes of models for classification

- discriminative models: e.g., logistic regression where $P(y = 1|x) = g(\theta^T x)$ and parameter θ is estimated by minimizing the cross-entropy J function.
- generative models: e.g., GDA and Naive Bayes where the joint distribution $P(xy)$ is estimated in terms of components $P(y)$ and $P(x|y)$ with appropriate parametric forms.

Both models yield linear separating hyperplanes of the form $\theta^T x = 0$ for a parameter vector θ.

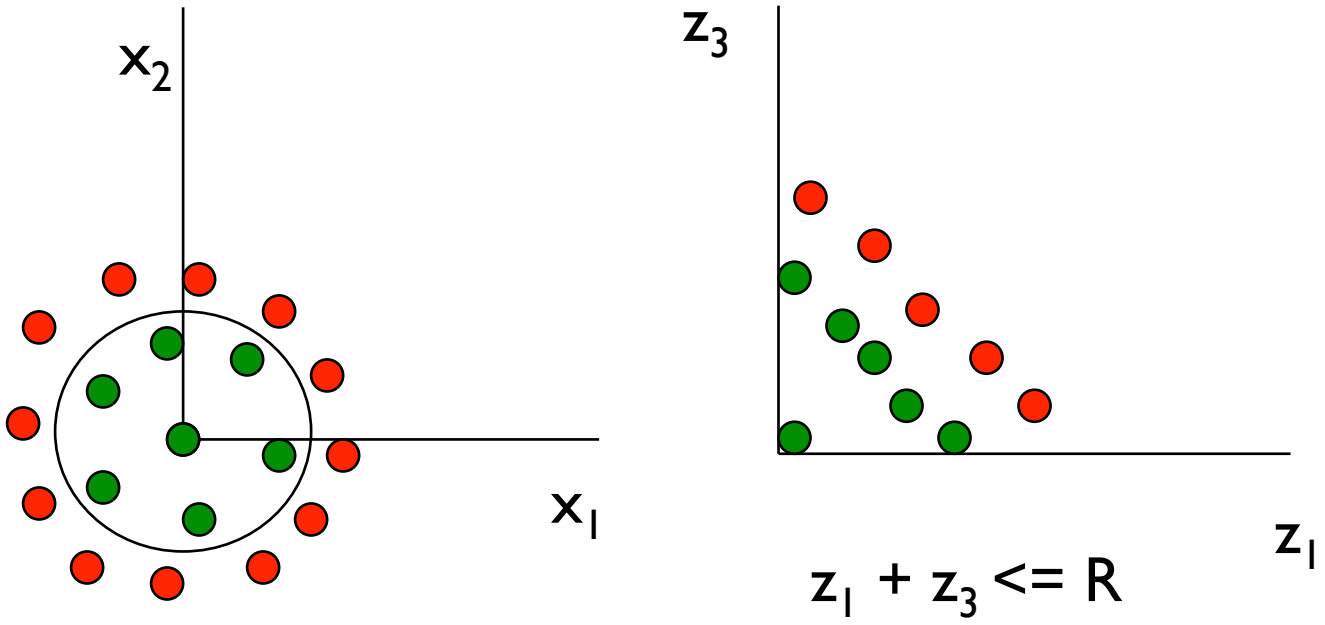
Can these two classes be separated?



Yes!

We use expanded basis functions and map each (x_1, x_2) pair to (x_1^2, x_2^2).

$$\phi((x_1, x_2)) = (x_1^2, x_2^2) = (z_1, z_3)$$



Now, the linear separating hyperplane can be characterized by $\theta^T \phi(x) = 0$.

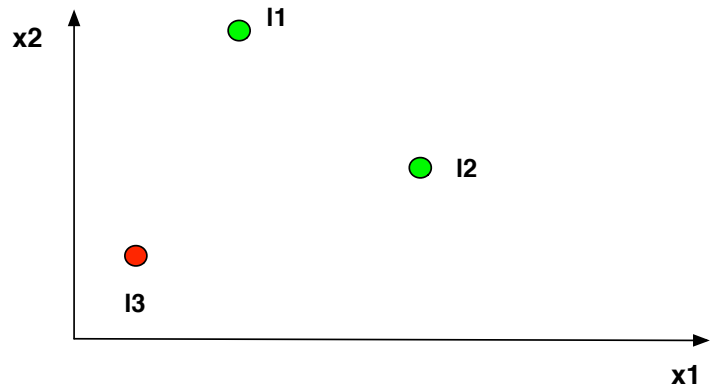
Approaches to constructing nonlinear classifiers

Nonlinear separating boundaries can be learned by linear models, but it places the burden of defining appropriate ϕ functions on the human.

There are two approaches to constructing nonlinear classifiers without explicit construction of basis functions.

- Compute classifications based on similarity between examples: kernel methods.
- Construct models by chaining together or layering simpler learning models: decision trees, neural networks, ensemble models.

Kernel methods: the intuition



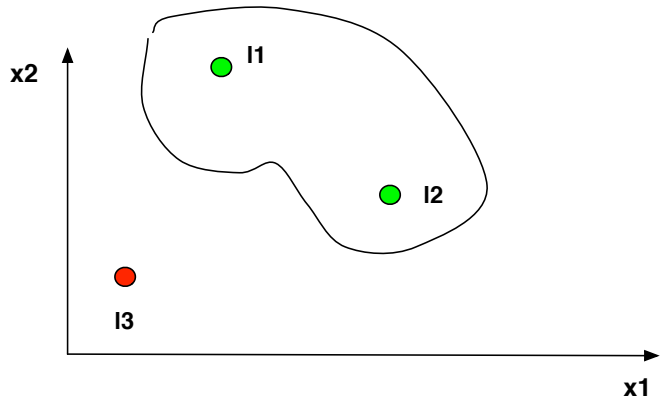
Let the input space be $\Re^2$ and let $l^{(1)}$, $l^{(2)}$, and $l^{(3)}$ be three labeled landmark points in that space. We define a classifier $h_\theta(x)$ parameterized by θ as follows:

$$h_\theta(x) = \theta_0 + \theta_1 similarity(x, l^{(1)}) + \theta_2 similarity(x, l^{(2)}) + \theta_3 similarity(x, l^{(3)})$$

Note that the new "features" are defined not just on x alone, but on the relationship between x and the three landmarks.

The decision rule for classification is: if $h_\theta(x) \geq 0$ then $y = 1$ else $y = 0$.

Kernel methods: an example



$$h_\theta(x) = -0.5 + 1 * similarity(x, l^{(1)}) + 1 * similarity(x, l^{(2)}) + 0 * similarity(x, l^{(3)})$$

$$similarity(x, l) = exp\left(-\frac{||x - l||^2}{2\sigma^2}\right) \quad \text{(Gaussian kernel)}$$

Key idea: we can form complex boundaries with features that are based on similarities between x and landmarks.

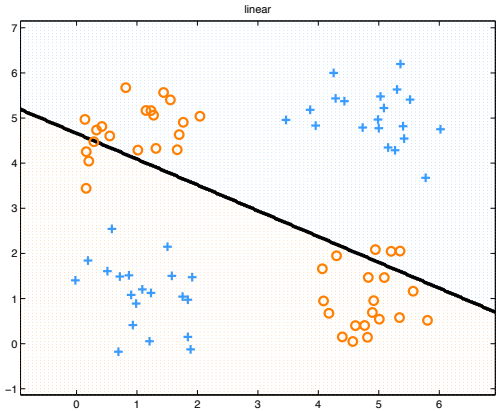
How to use kernels for classification

Given $\mathcal{D} = \{(x^{(i)}, y^{(i)}) | 1 \leq i \leq m, x^{(i)} \in \Re^d, y^{(i)} \in \{0, 1\}\}$, choose all the points in $\mathcal{D}$ as landmarks. Then, represent each $x^{(i)}$ by a feature vector of length m as follows:

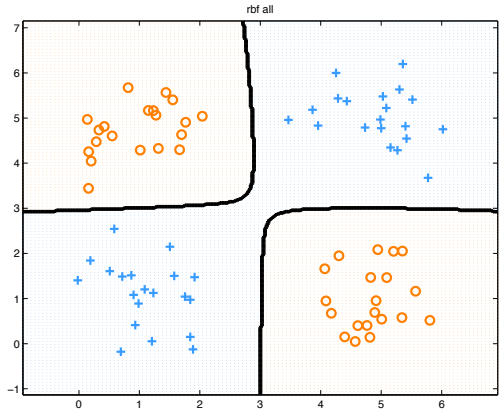
$$x^{(i)} \rightarrow \begin{bmatrix} sim(x^{(i)}, x^{(1)}) \\ \dots \\ sim(x^{(i)}, x^{(m)}) \end{bmatrix} = fx^{(i)}, fX = \begin{bmatrix} 1 & \text{———} fx^{(1)\,T} \text{———} \\ 1 & \text{———} fx^{(2)\,T} \text{———} \\ \dots & \\ 1 & \text{———} fx^{(m)\,T} \text{———} \end{bmatrix}$$

Now we can use regularized logistic regression in the usual way to estimate $\theta \in \Re^{m+1}$. Given a new example x, we can compute $g(\theta^T fx)$ to predict the class associated with x.

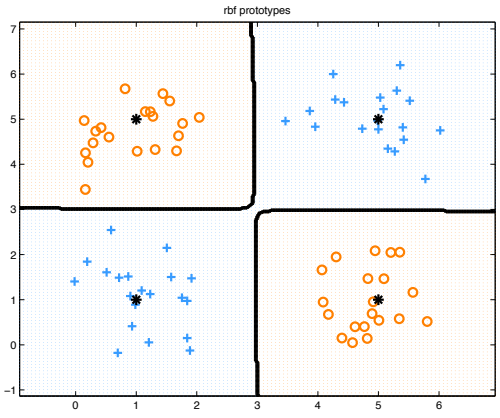
Example: kernelized logistic regression



logistic



all points are landmarks



4 points are landmarks

How to pick landmarks

- Use all examples in $\mathcal{D}$ as landmarks and use aggressive regularization, especially if m is large.
- Cluster the examples in $\mathcal{D}$ and choose cluster centers as landmarks.
- if the input space is $\Re^d$ where d is small, choose landmarks that tile $\Re^d$ uniformly.

Similarity/kernel functions

A similarity or kernel function k measures how similar two x's from $\Re^d$ are.

$$k : \Re^d \times \Re^d \rightarrow \Re$$

Examples of kernel functions

- Gaussian kernel

$$k(x, x') = exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

- Second-order polynomial kernel

$$k(x, x') = (x^T x')^2$$

Mercer's theorem

Mercer's theorem provides a necessary and sufficient condition for a function $k(x, x^{'})$ to be a valid kernel. The $m \times m$ Gram matrix K whose elements are $k(x^{(i)}, x^{(j)})$, $1 \leq i, j \leq m$ should be positive definite for all choices of the set $\{x^{(i)} : 1 \leq i \leq m\}$.

If K is positive definite, then there exists a basis function ϕ such that every entry $k(x^{(i)}, x^{(j)})$ in K can be written as

$$k(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T \phi(x^{(j)})$$

By using the kernel function k we bypass the construction of ϕ. For example, if $d = 16 \times 16$ and we consider all fifth-order polynomial terms on the original feature space, we would have to construct a feature space of size 10^{10} if we explicitly constructed the basis function ϕ. Instead, we can use a 5th order polynomial kernel and compute $(x^T x^{'})^5$ in $O(d)$ time.