Learning From Data
Lecture 10
Nonlinear Transforms

The Z-space
Polynomial transforms
Be careful

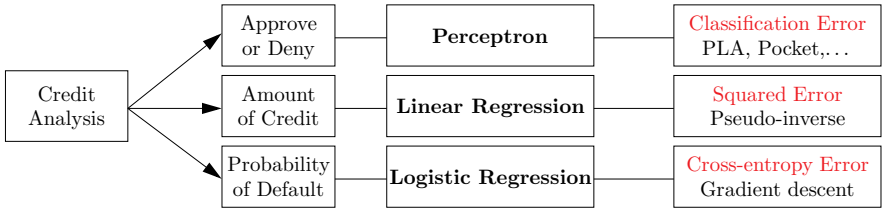
M. Magdon-Ismail
CSCI 4100/6100

RECAP: The Linear Model

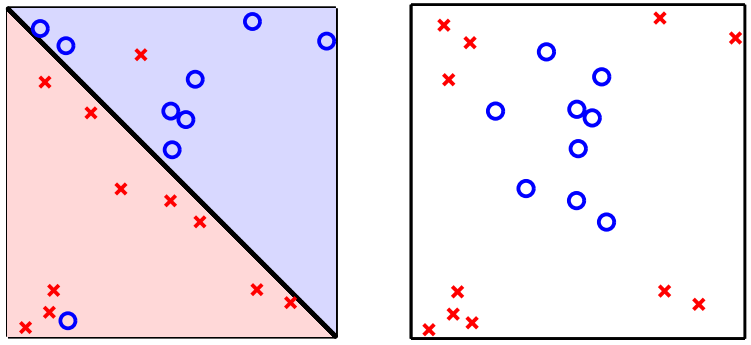
linear in $\mathbf{x}$: gives the line/hyperplane separator

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

linear in $\mathbf{w}$: makes the algorithms work



The Linear Model has its Limits

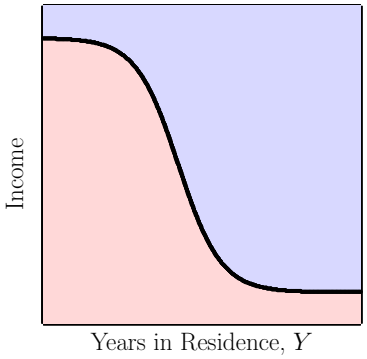


(a) Linear with outliers

(b) Essentially nonlinear

To address (b) we need something more than linear.

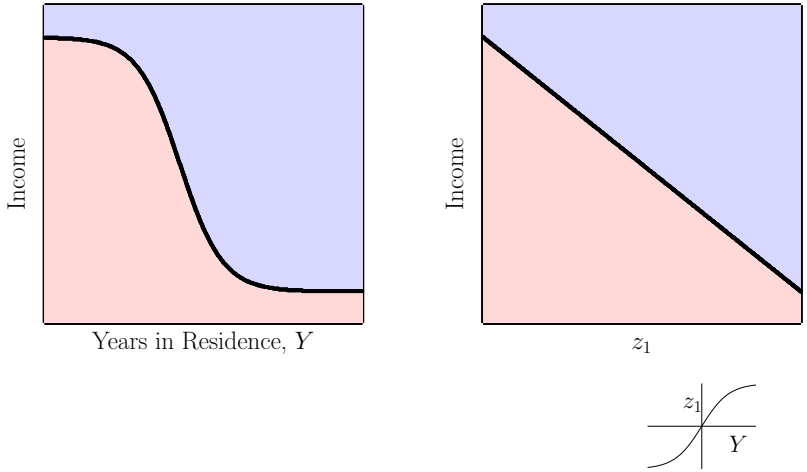
Change Your Features



$Y \gg 3$ years
no additional effect beyond $Y = 3$;

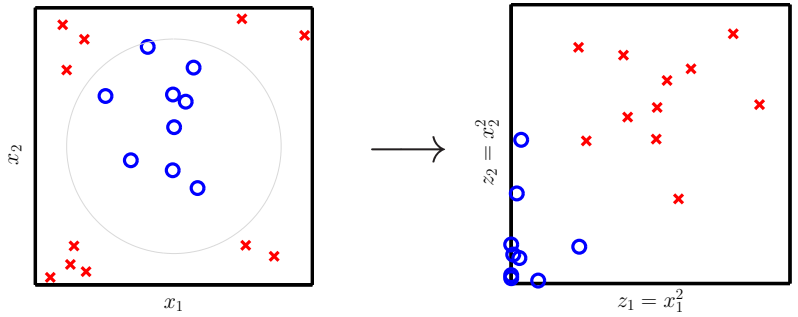
$Y \ll 0.3$ years
no additional effect below $Y = 0.3$.

Change Your Features Using a Transform



Mechanics of the Feature Transform I

Transform the data to a $\mathcal{Z}$-space in which the data is separable.

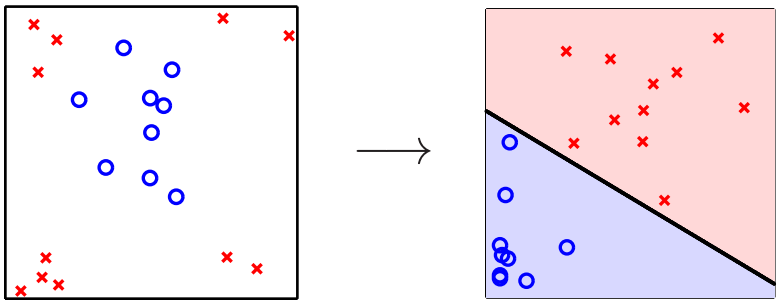


$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \longrightarrow \mathbf{z} = \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ x_1^2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \Phi_2(\mathbf{x}) \end{bmatrix}$$

Mechanics of the Feature Transform II

Separate the data in the $\mathcal{Z}$-space with $\tilde{\mathbf{w}}$:

$$\tilde{g}(\mathbf{z}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z})$$

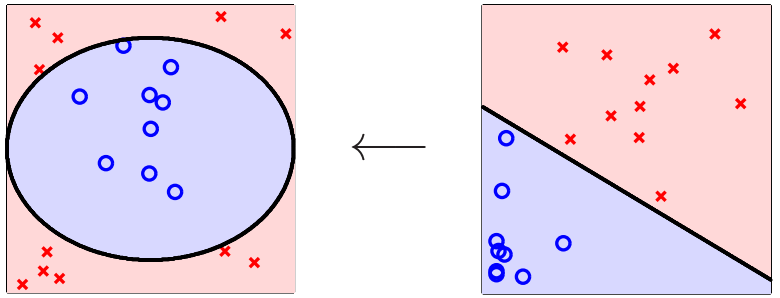


Mechanics of the Feature Transform III

To classify a new $\mathbf{x}$, first transform $\mathbf{x}$ to $\mathbf{\Phi}(\mathbf{x}) \in \mathcal{Z}$-space and classify there with $\tilde{g}$.

$$\begin{aligned} g(\mathbf{x}) &= \tilde{g}(\mathbf{\Phi}(\mathbf{x})) \\ &= \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x})) \end{aligned}$$

$$\tilde{g}(\mathbf{z}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z})$$



The General Feature Transform

$\mathcal{X}$-space is $\mathbb{R}^d$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$

$y_1, y_2, \ldots, y_N$

no weights

$\mathcal{Z}$-space is $\mathbb{R}^{\tilde{d}}$

$$\mathbf{z} = \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \vdots \\ \Phi_{\tilde{d}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_{\tilde{d}} \end{bmatrix}$$

$\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$

$y_1, y_2, \ldots, y_N$

$$\tilde{\mathbf{w}} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{d}} \end{bmatrix}$$

$g(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}))$

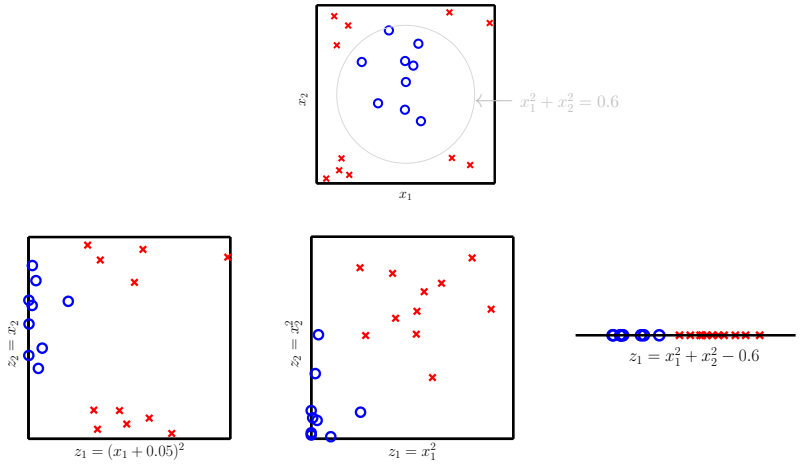
Generalization

d_{VC} $\qquad$ $\tilde{d}_{\text{VC}}$

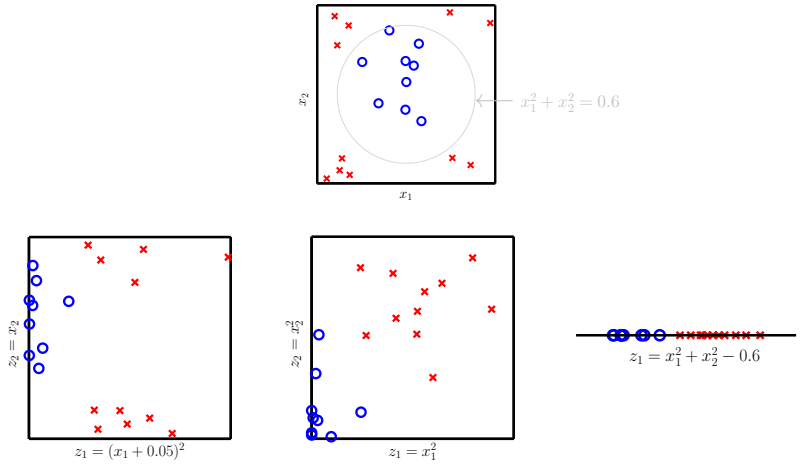
$d + 1 \quad \longrightarrow \quad \boldsymbol{\tilde{d} + 1}$

Choose the feature transform with smallest $\tilde{d}$

Many Nonlinear Features May Work



Many Nonlinear Features May Work



A rat! A rat!

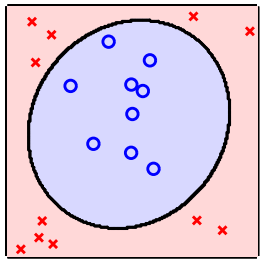
This is called data snooping: looking at your data and tailoring your $\mathcal{H}$.

Must Choose Φ BEFORE Your Look at the Data

After constructing features carefully, **before** seeing the data . . .

. . . if you think linear is not enough, try **the 2nd order polynomial transform**.

$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \mathbf{x} \longrightarrow \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \Phi_2(\mathbf{x}) \\ \Phi_3(\mathbf{x}) \\ \Phi_4(\mathbf{x}) \\ \Phi_5(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}$$



The General Polynomial Transform Φ_k

We can get even fancier: degree-k polynomial transform:

$$\begin{aligned}
\mathbf{\Phi}_1(\mathbf{x}) &= (1, x_1, x_2), \\
\mathbf{\Phi}_2(\mathbf{x}) &= (1, x_1, x_2, x_1^2, x_1x_2, x_2^2), \\
\mathbf{\Phi}_3(\mathbf{x}) &= (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3), \\
\mathbf{\Phi}_4(\mathbf{x}) &= (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4), \\
&\vdots
\end{aligned}$$

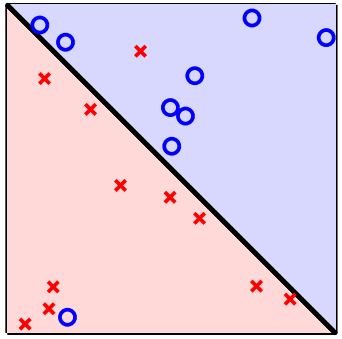
– Dimensionality of the feature space increases rapidly (d_{VC})!

– Similar transforms for d-dimensional original space.

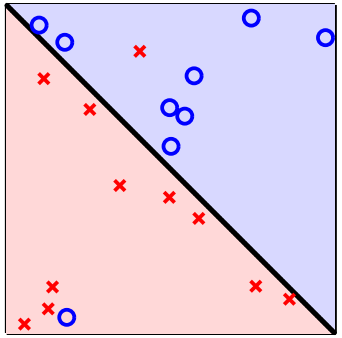
– Approximation-generalization tradeoff

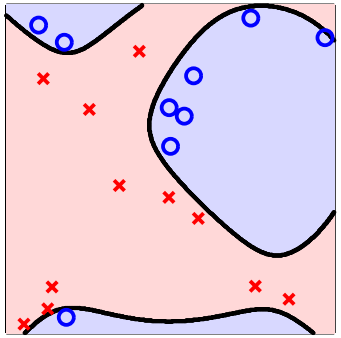
Higher degree gives lower (even zero) E_{in} but worse generalization.

Be Careful with Feature Transforms



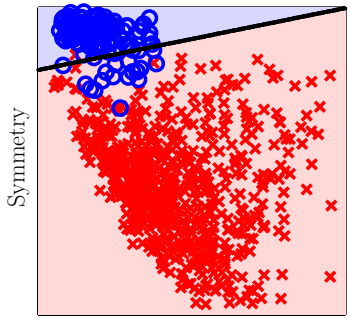
Be Careful with Feature Transforms



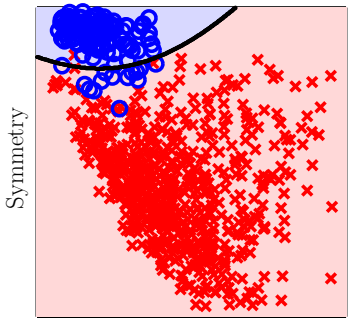


High order polynomial transform leads to "nonsense".

Digits Data "1" Versus "All"



Linear model
$E_{\text{in}} = 2.13\%$
$E_{\text{out}} = 2.38\%$



3rd order polynomial model
$E_{\text{in}} = 1.75\%$
$E_{\text{out}} = 1.87\%$

Use the Linear Model!

- First try a linear model – simple, robust and works.
- Algorithms can tolerate error plus you have nonlinear feature transforms.
- Choose a feature transform *before* seeing the data. Stay simple.
  Data snooping is hazardous to your E_{out}.
- Linear models are fundamental in their own right; they are also the building blocks of many more complex models like support vector machines.
- Nonlinear transforms also apply to regression and logistic regression.