

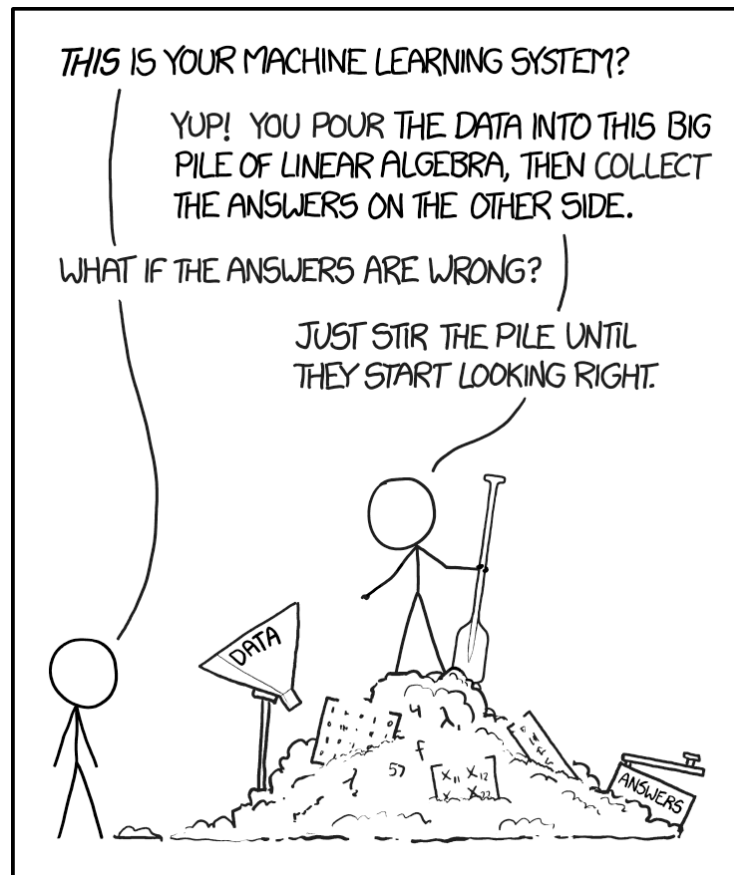
# Summary

Pascal Fua  
IC-CVLab

# What is Machine Learning?

Machine learning algorithms:

- seek to provide knowledge to computers through data, observations, and interaction with the world,
- can make predictions given new observations,
- improve when given access to large amounts of training data.



# Artificial Intelligence

## Artificial Intelligence

*Expert Systems*

*A\**

*min-max*

## Machine Learning

*Support Vector Machines*

*Boosting*

*Random Forests*

## Deep Learning

*Lenet*

*VGG*

*ResNet*



1997: Deep blue beats chess world champion



2017: AlphaGo beats go world champion

# Self-Driving Cars



1985  
DARPA ALV



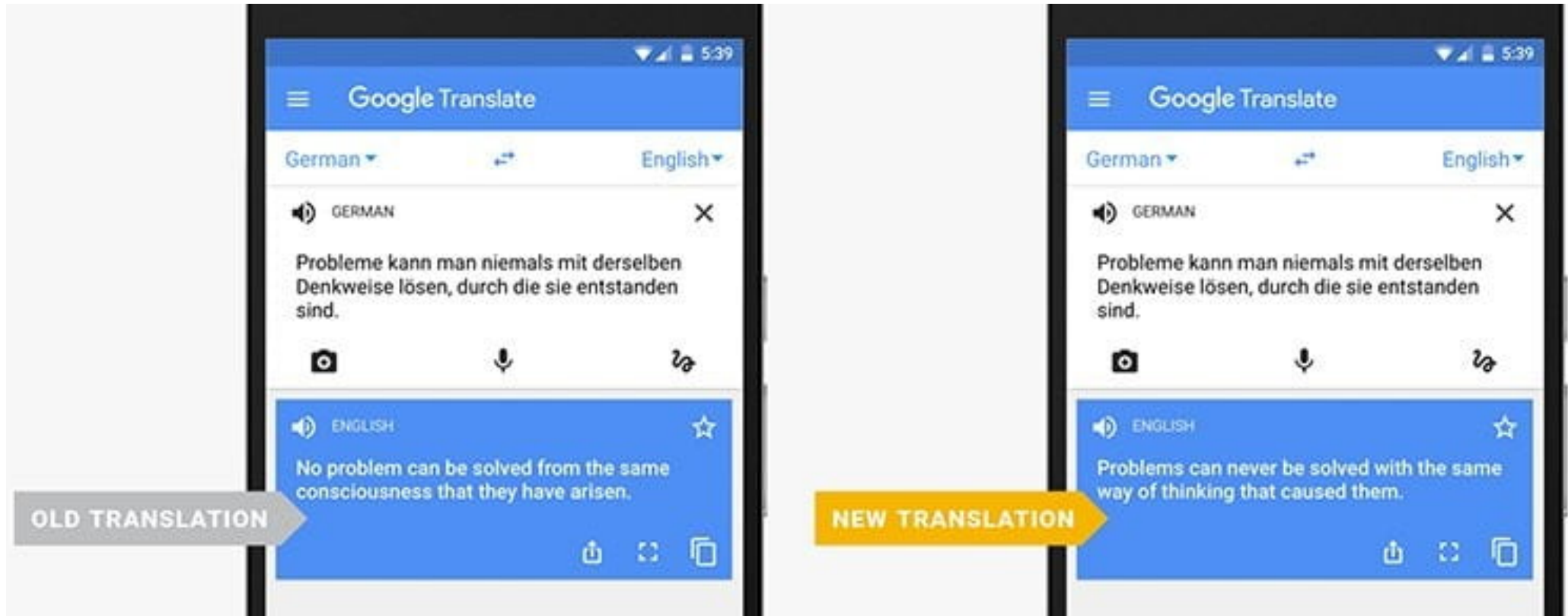
2007  
DARPA Urban Challenge



2010  
Google  
Tesla

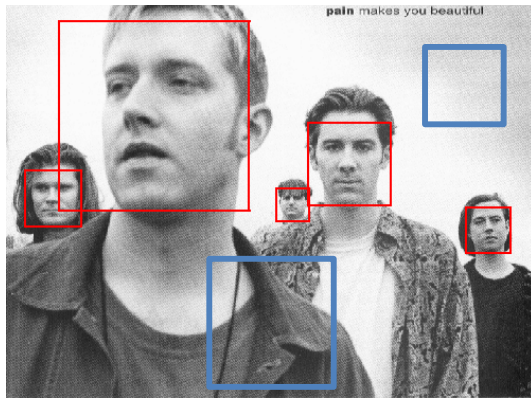
- More computing power.
- Better sensors.
- Detailed maps of the environment.
- **Machine learning**

# Machine Translation



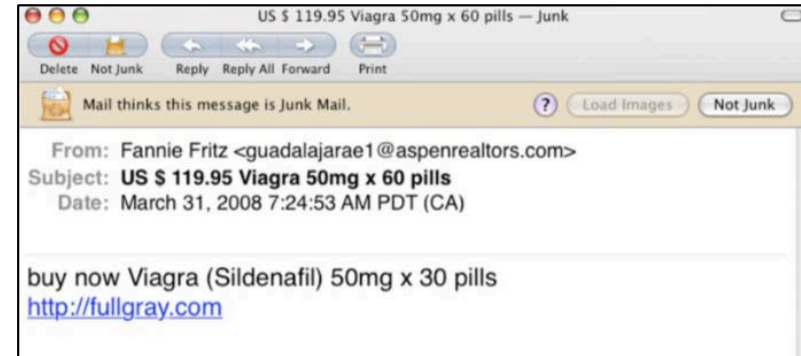


# Classification vs Regression

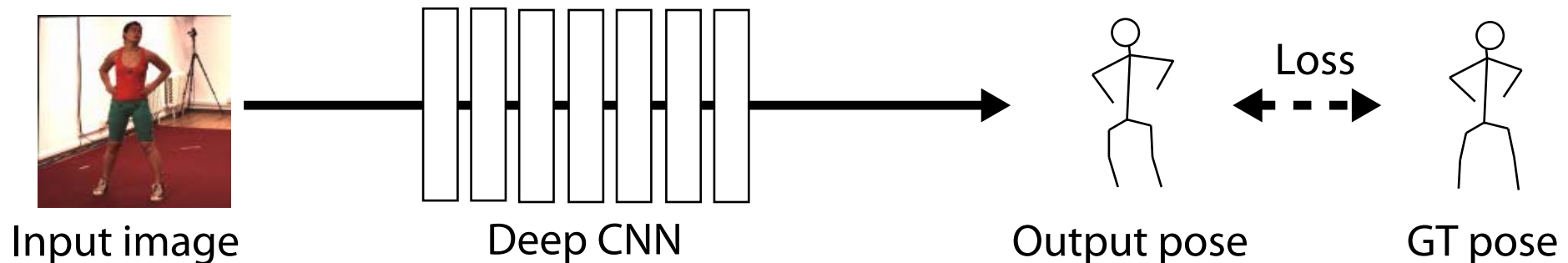


Face  
 Not Face

Spam or not



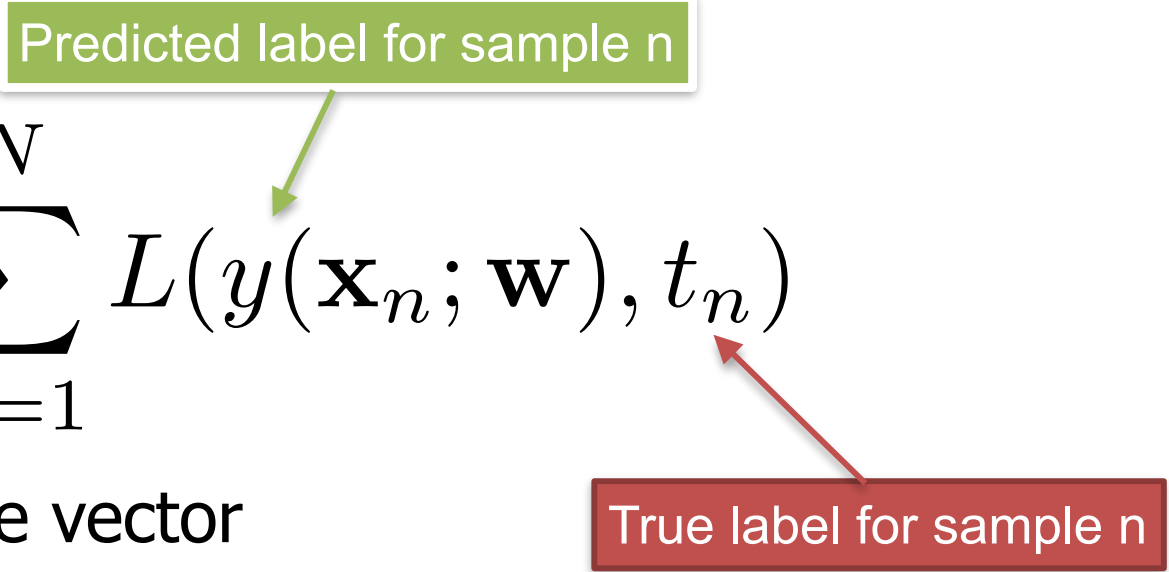
- Classification algorithms seek to estimate a mapping function  $y$  from the input vector  $\mathbf{x}$  to discrete or categorical output variables.



- Regression algorithms seek to estimate the mapping function  $y$  from the input vector  $\mathbf{x}$  to **numerical or continuous** output variables.

# Supervised Classification

Minimize:

$$E(\mathbf{w}) = \sum_{n=1}^N L(y(\mathbf{x}_n; \mathbf{w}), t_n)$$


- **x**: Feature vector
- **w**: Model parameters
- **t**: Label
- **y**: Predictor
- **L**: Loss Function
- **E**: Error Function

—> ML is an optimization problem

# Classification Techniques

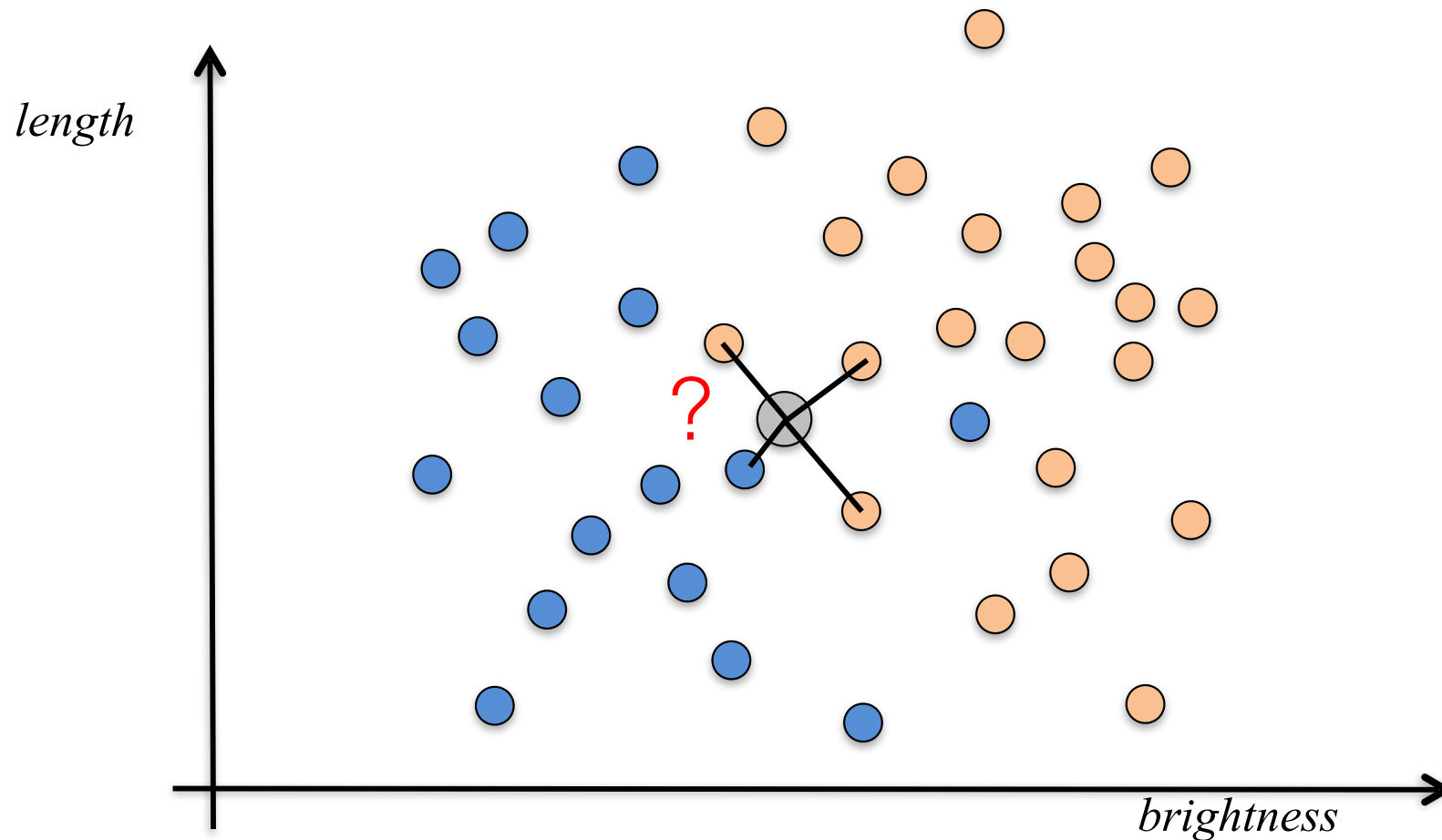
- Nearest-Neighbors and K-Means
- Logistic Regression
- Boosting
- Support Vector Machines
- Decision Trees and Forests
- Multilayer Perceptrons
- Convolutional Neural Networks



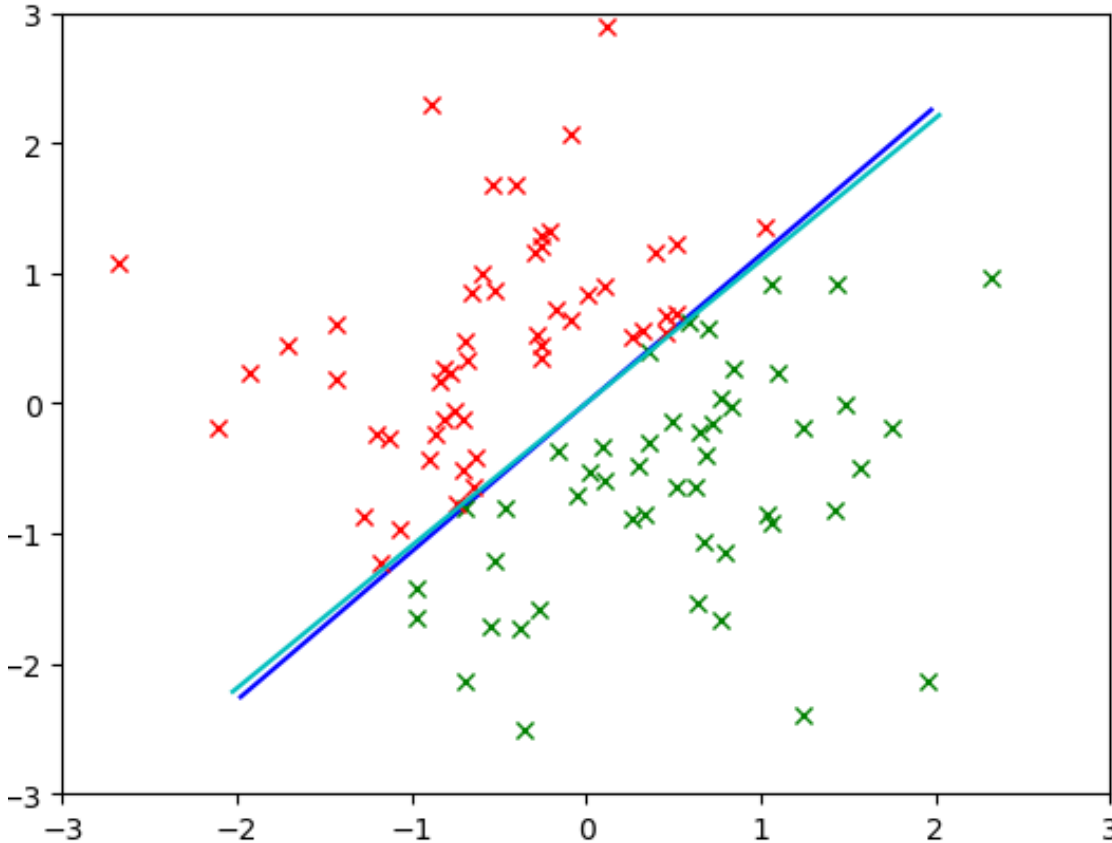
# K-Nearest-Neighbor Classifier

Improved algorithm:

- Given a new  $\mathbf{x}$  to be classified, find its k nearest neighbors in the training set.
- Classify the point according to the majority of labels of its nearest neighbors.



# Perceptron

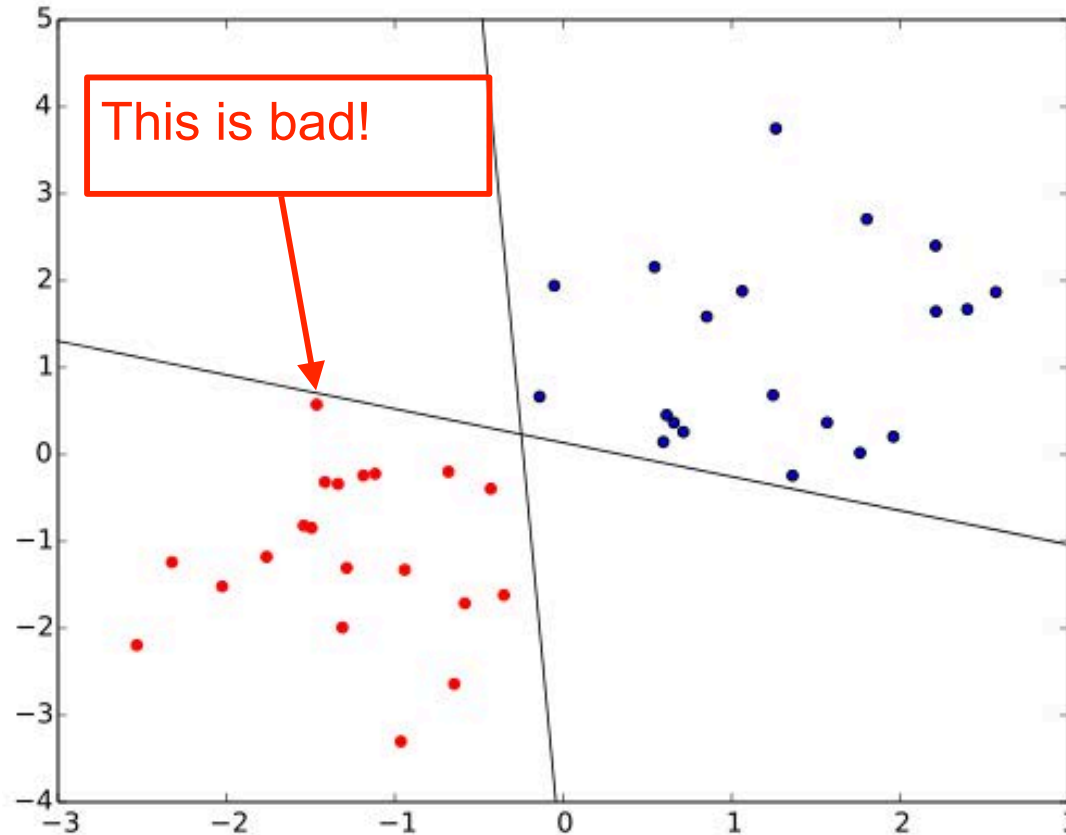


$$y(\mathbf{x}; \mathbf{w}, w_0) = \begin{cases} 1 & \text{if } \mathbf{w}^T \mathbf{x} + w_0 \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Given the training set  $\{(x_n, t_n)_{1 \leq n \leq N}\}$ , choose a  $\mathbf{w}$  that minimizes

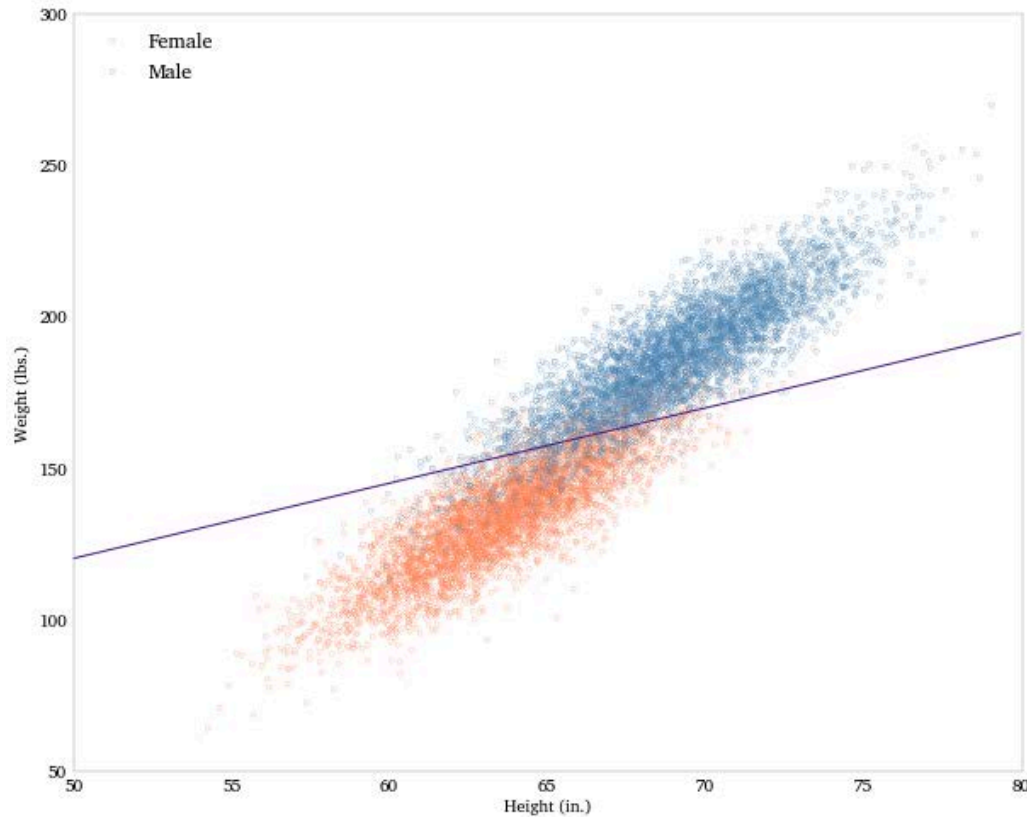
$$E(\mathbf{w}, w_0) = - \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0) t_n$$

# The Problem with the Perceptron

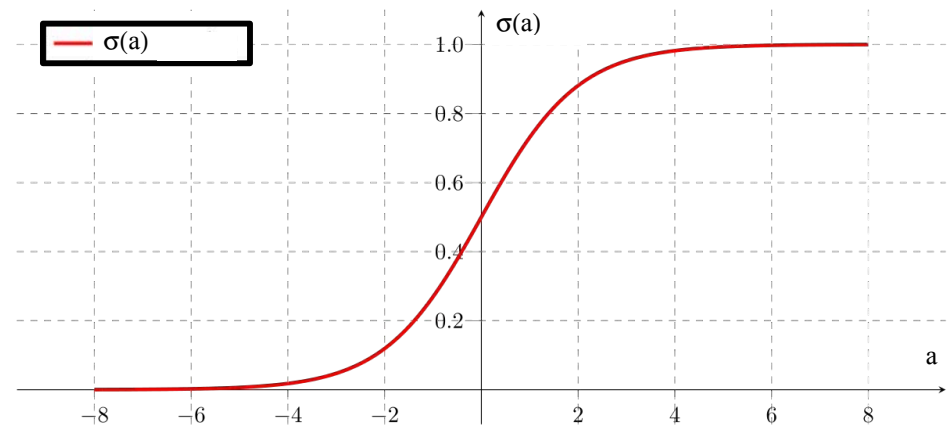


- Two different solutions among infinitely many.
- The perceptron has no way to favor one over the other.

# Logistic Regression



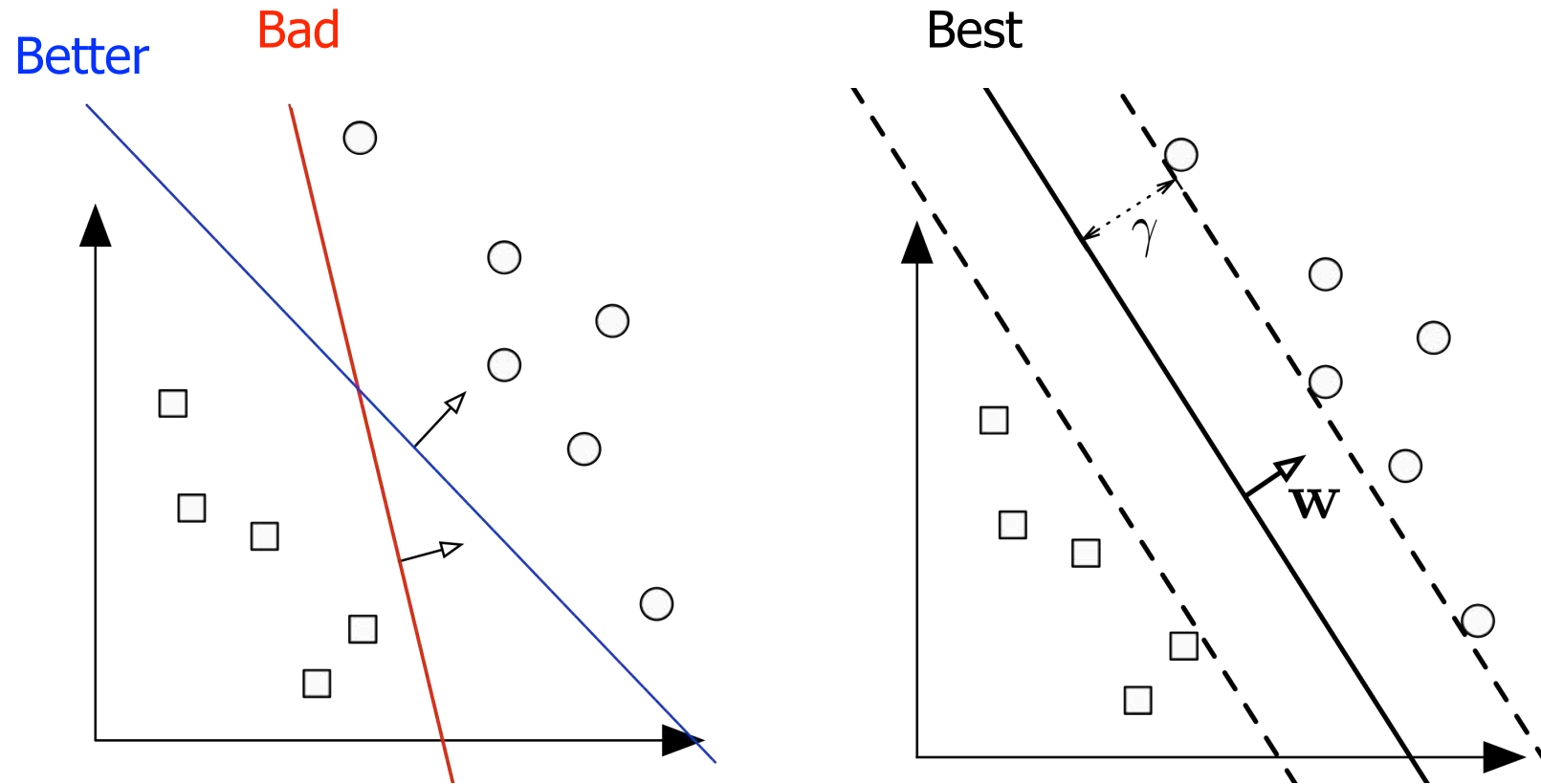
$$y(\mathbf{x}; \mathbf{w}, w_0) = \sigma(\mathbf{w}^T \mathbf{x} + w_0) \\ \approx p(t = 1, \mathbf{x})$$



Given the training set  $\{(x_n, t_n)_{1 \leq n \leq N}\}$ , choose a  $\mathbf{w}$  that minimizes

$$E(\mathbf{w}, w_0) = - \sum_n \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \approx - \ln(p(\mathbf{t} | \mathbf{w}, w_0)) .$$

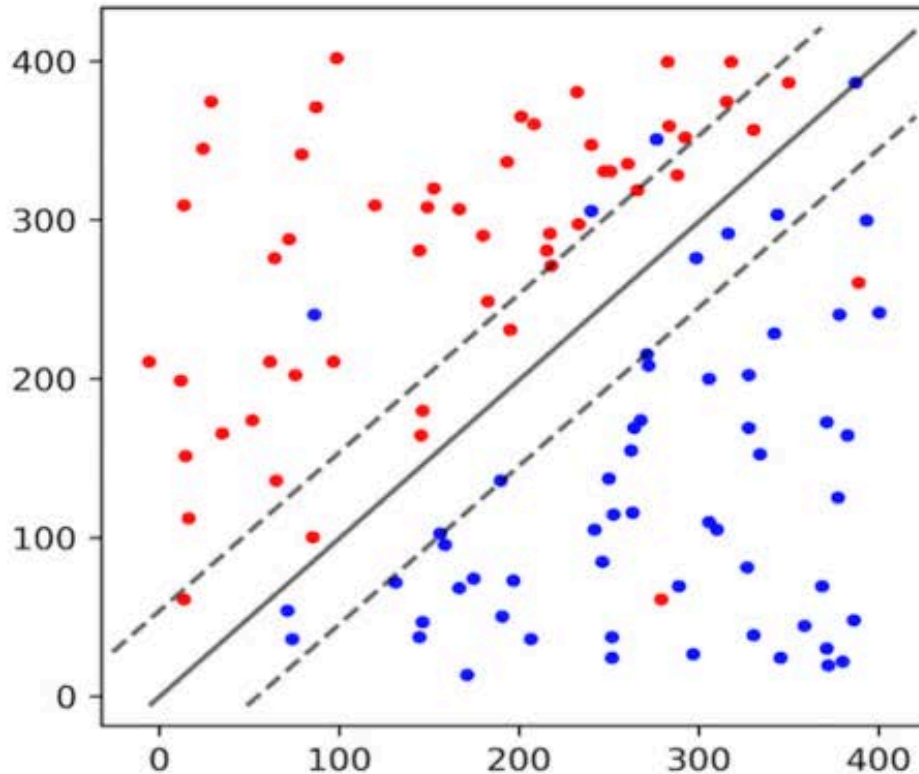
# Maximizing the Margin



- The larger the margin, the better!
- In the presence of outliers, the logistic regression does not guarantee a large one.

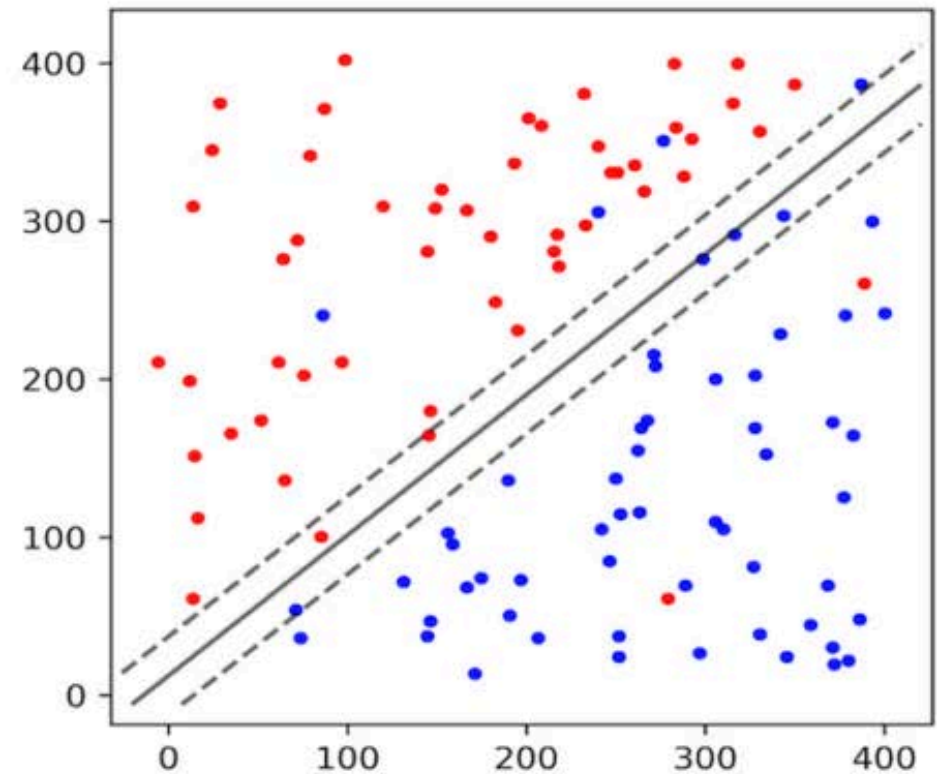
How do we maximize it?

# Max Margin Classifier



$C=1$ :

- Large margin.
- Many training samples misclassified.



$C=100$ :

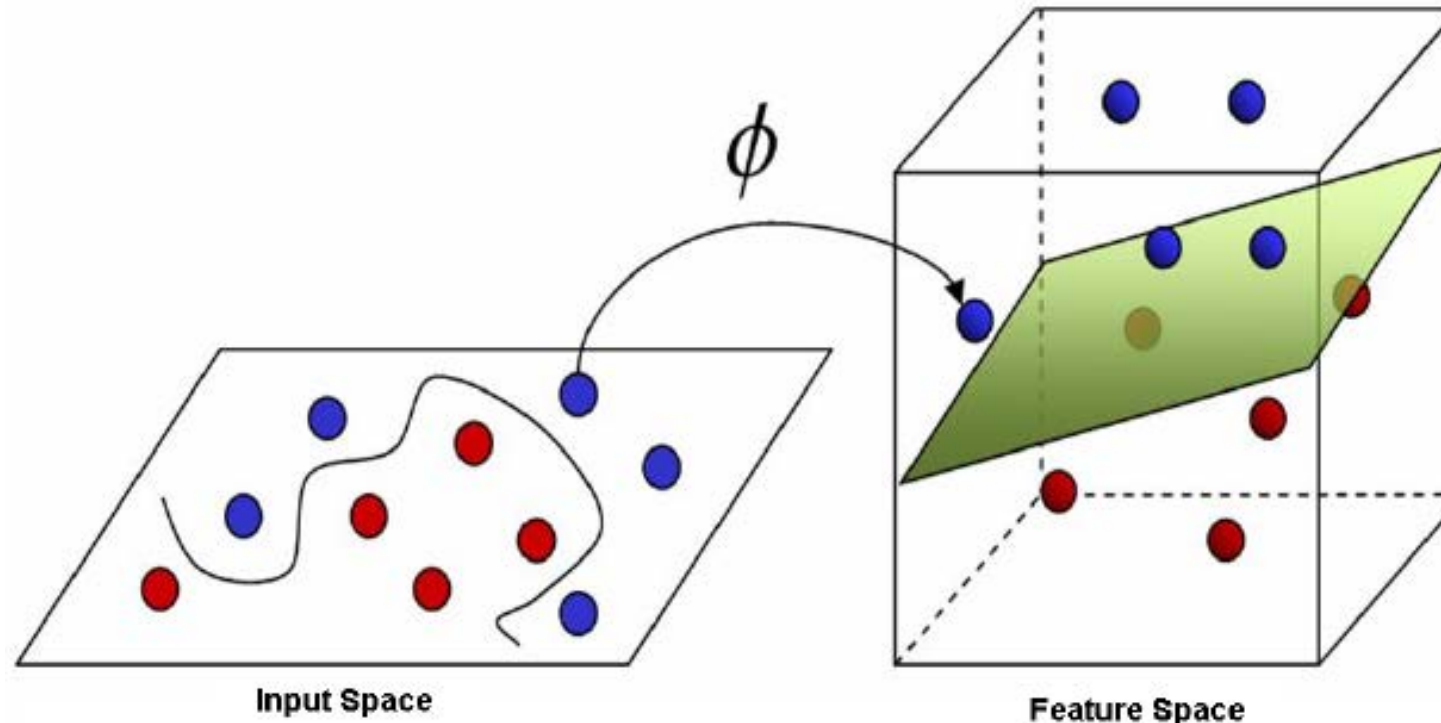
- Small margin.
- Few training samples misclassified.

Which is best?

- It depends.
- Must use cross-validation, as we did for k-Means.

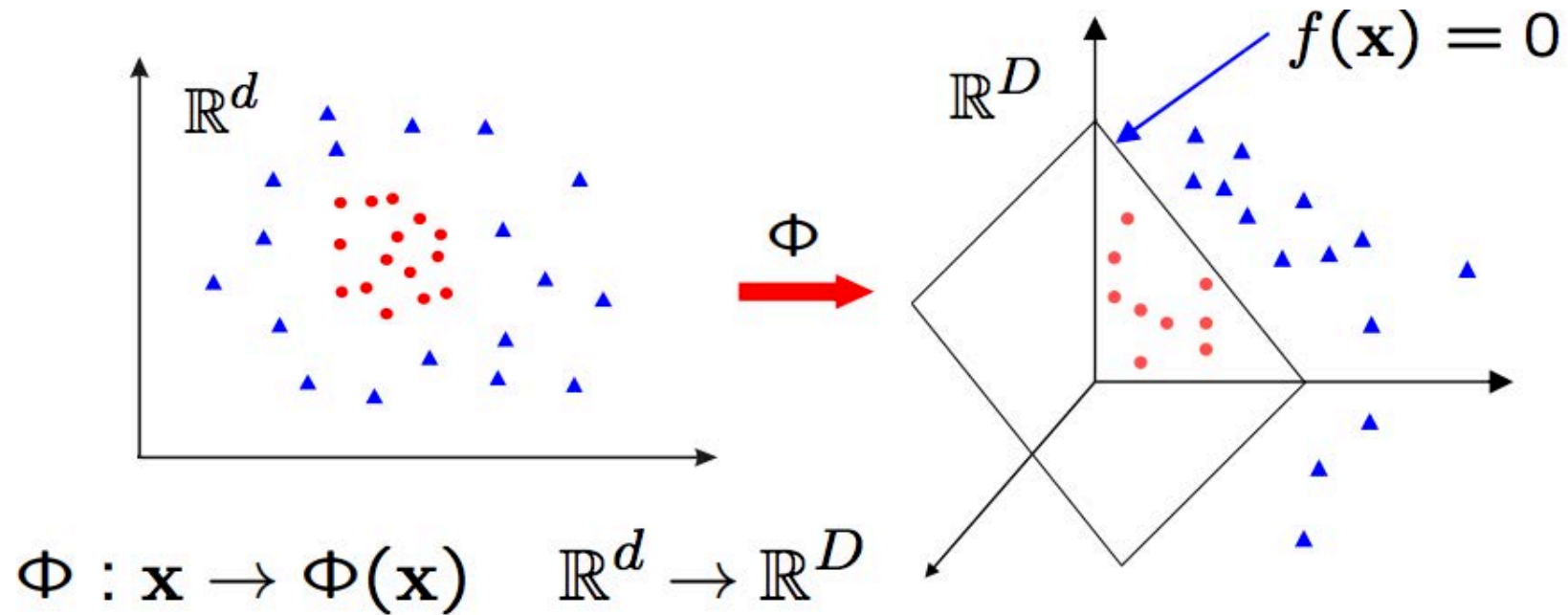


# Non Linearly Separable Data



- Map to a higher dimensional space in which it is.
- Use an ensemble of classifiers.
- Use a deep network.

# Classification in Feature Space



- Map from  $\mathbb{R}^d$  to  $\mathbb{R}^D$
- Learn a linear classifier in  $\mathbb{R}^D$

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}) + w_0)$$

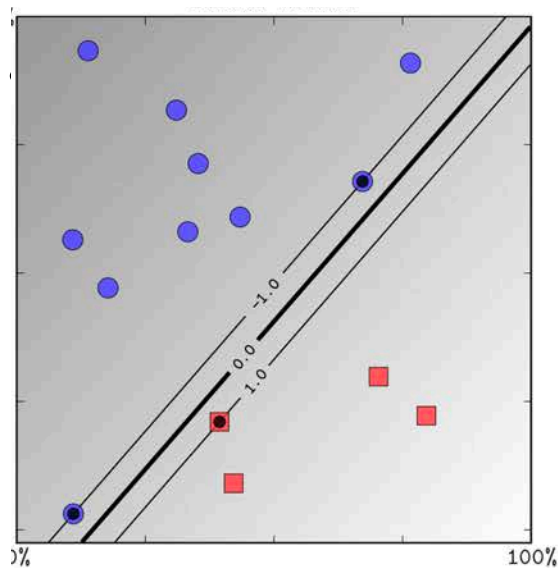
$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

# Polynomial SVMs

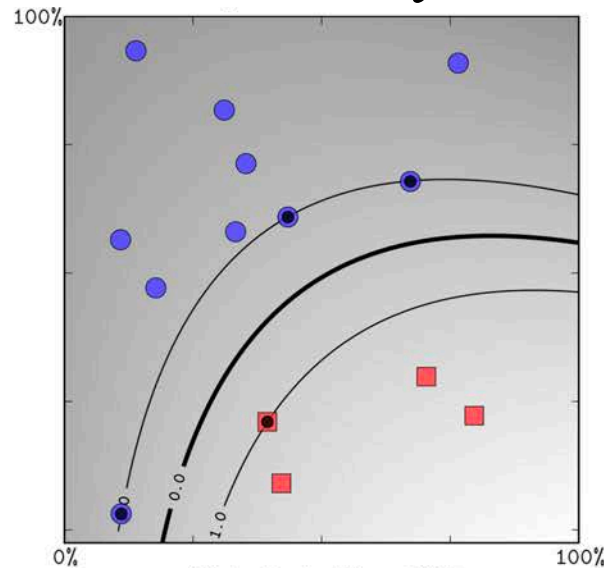
$$\mathbf{w}^* = \min_{(\mathbf{w}, \{\xi_n\})} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n,$$

subject to  $\forall n, \quad t_n \cdot (\tilde{\mathbf{w}} \cdot \phi(\mathbf{x}_n)) \geq 1 - \xi_n$  and  $\xi_n \geq 0$ .

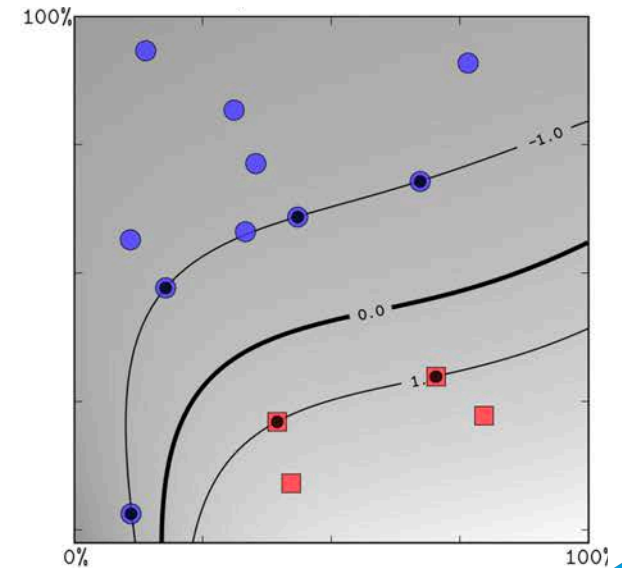
- $C$  is constant that controls how costly constraint violations are.



M = 1

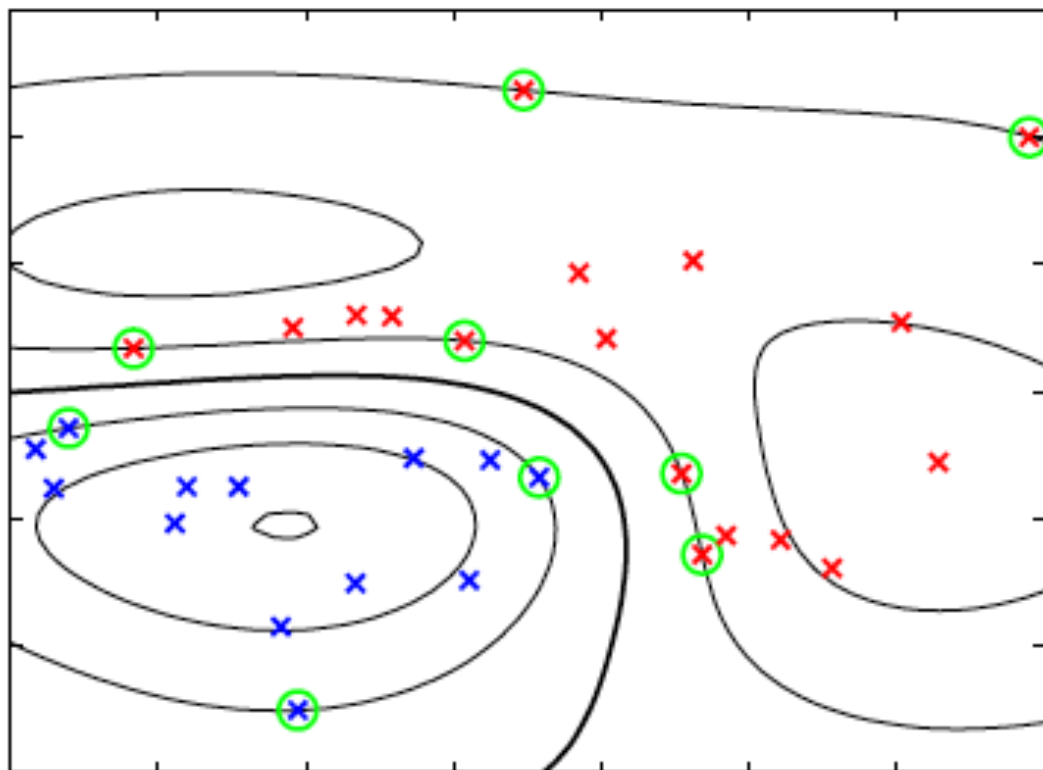


M = 2



M = 5

# Kernel SVMs



$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) .$$

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

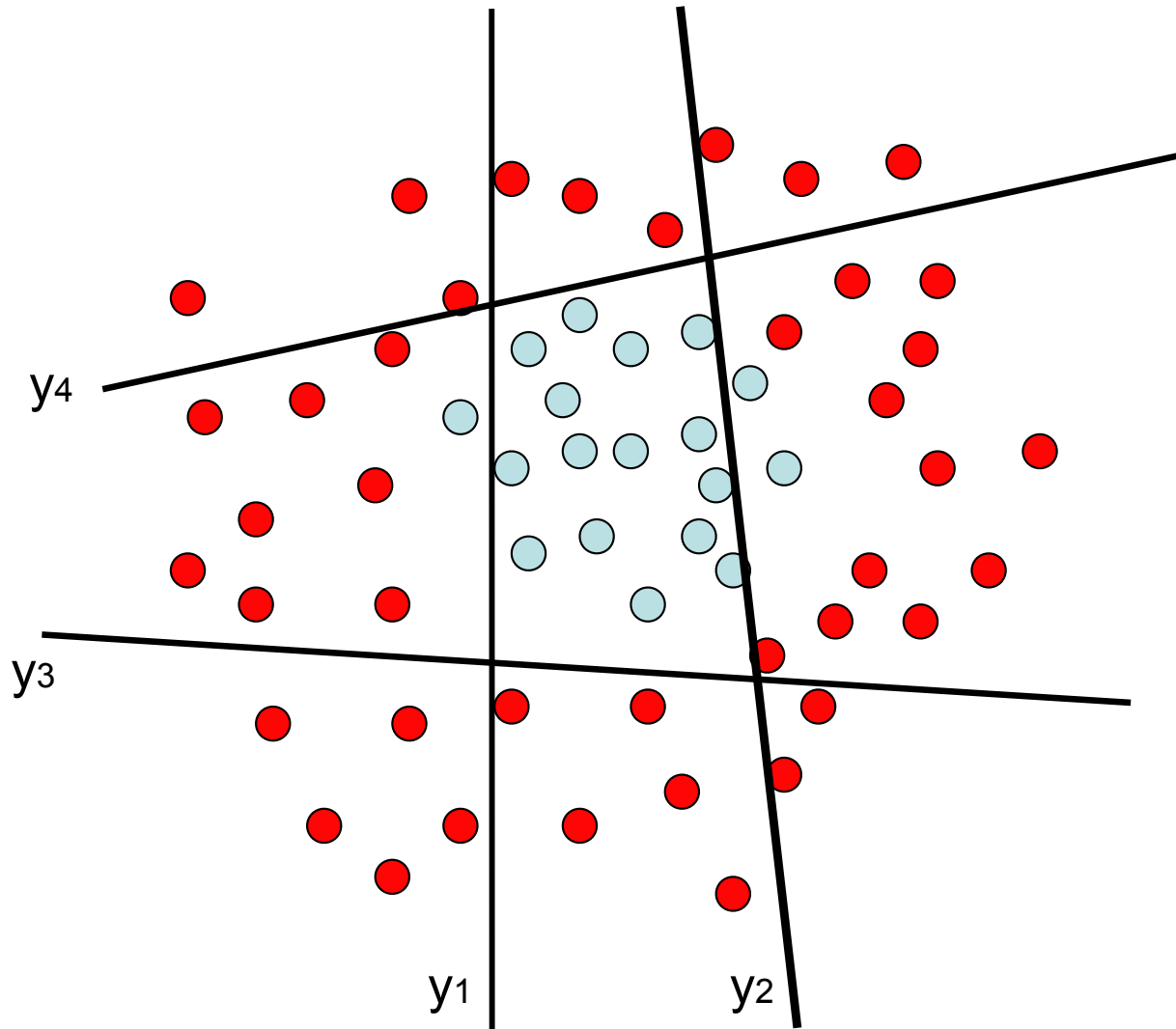
$$= \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b ,$$

$$\text{with } k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') .$$

- Only for a subset of the data points is  $a_n$  is non zero.
- The corresponding  $\mathbf{x}_n$  are the support vectors and satisfy  $t_n y(\mathbf{x}_n) = 1$ .
- They are the only ones that need to be considered as test time.

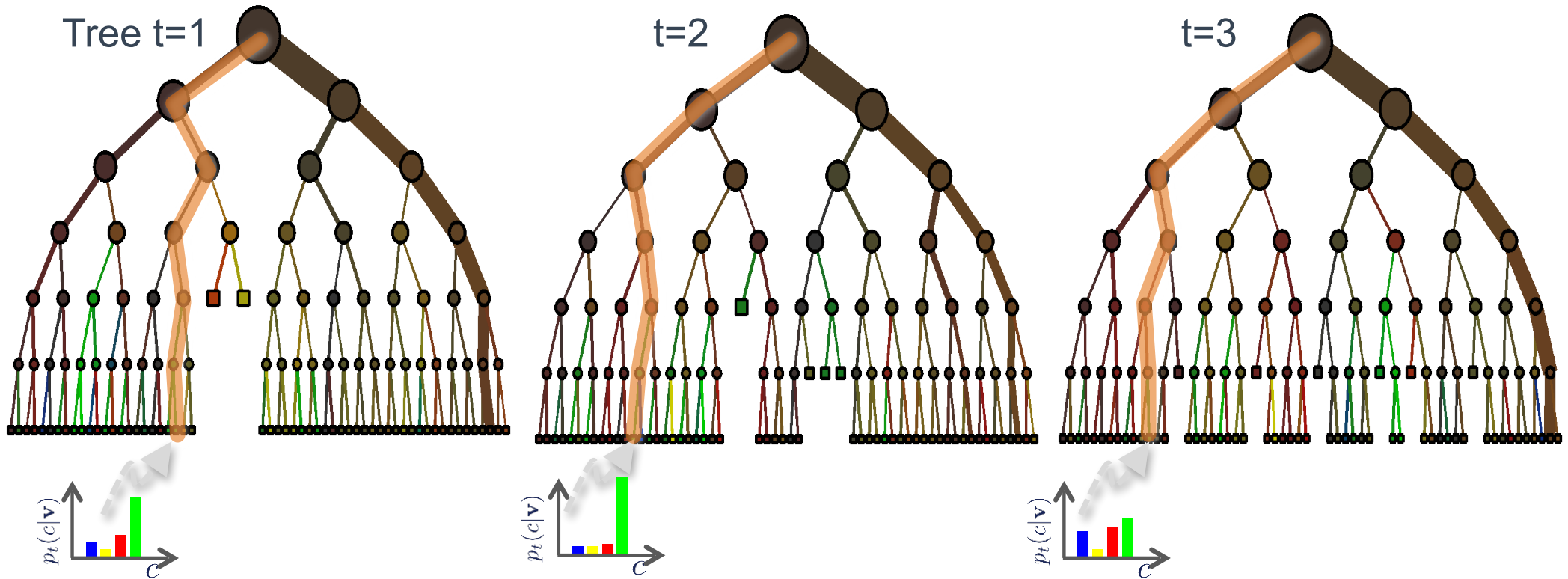
—> That is what makes SVMs practical!

# AdaBoost

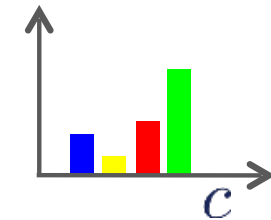


$$y(\mathbf{x}) = \alpha_1 y_1(\mathbf{x}) + \alpha_2 y_2(\mathbf{x}) + \alpha_3 y_3(\mathbf{x}) + \alpha_4 y_4(\mathbf{x})$$

# Decision Forests



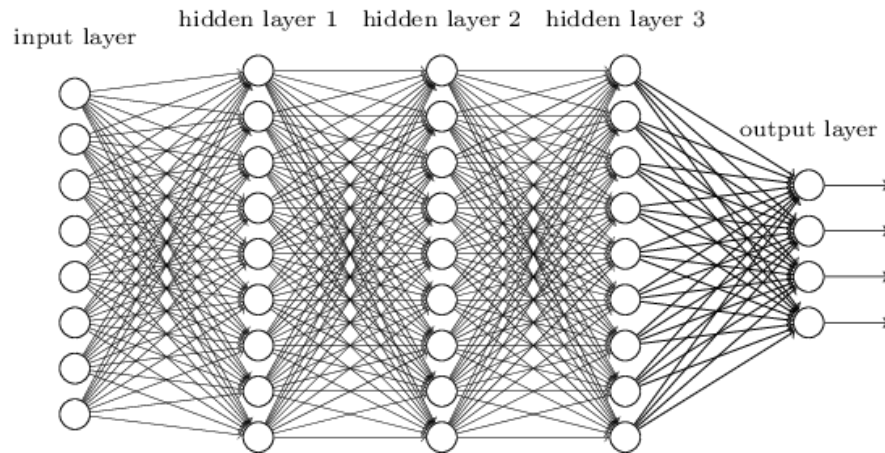
$$L(c, \mathbf{v}) = \frac{1}{T} \sum_t -\log(p_t(c|\mathbf{v}))$$



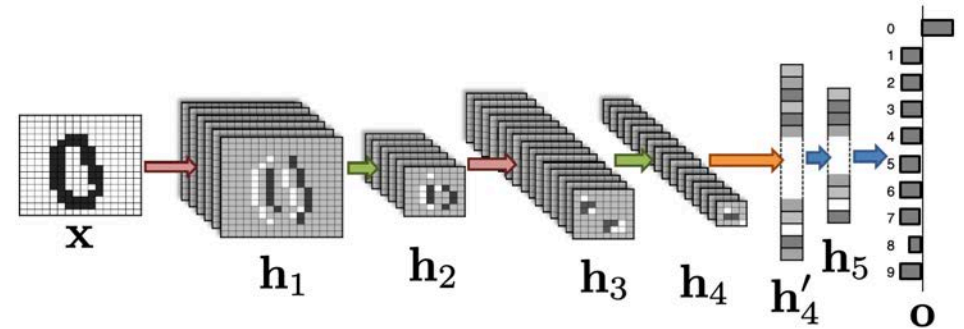
Simple and flexible approach.



# Neural Networks



Fully connected



Convolutional

- Some of the most powerful current techniques around when enough training data is available.
- Convolutional Neural Nets are particularly well adapted for image processing.

# Regression Techniques

- Linear Regression
- Polynomial Regression
- Neural Networks

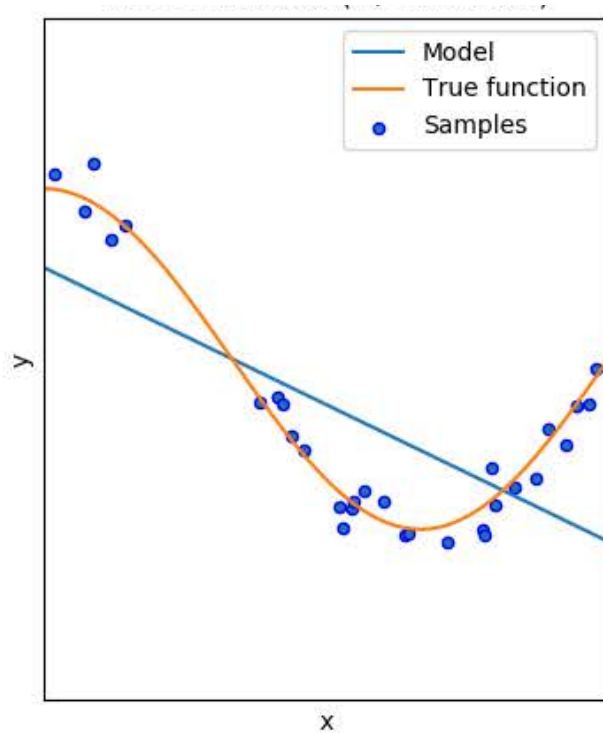
# Supervised Regression

$$\text{Minimize } E(\mathbf{w}) = \sum_{n=1}^N L(y(\mathbf{x}_n; \mathbf{w}), t_n)$$

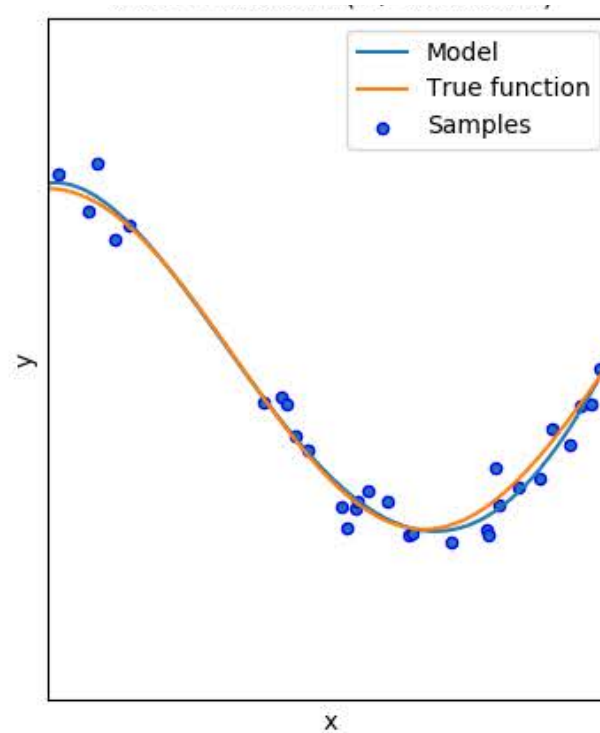
- **x**: Feature vector
- **w**: Model parameters
- **t**: Label
- **y**: Predictor
- **L**: Loss Function
- **E**: Error Function

Same as for classification, except for the fact that the  $t_n$  now denotes continuous values!

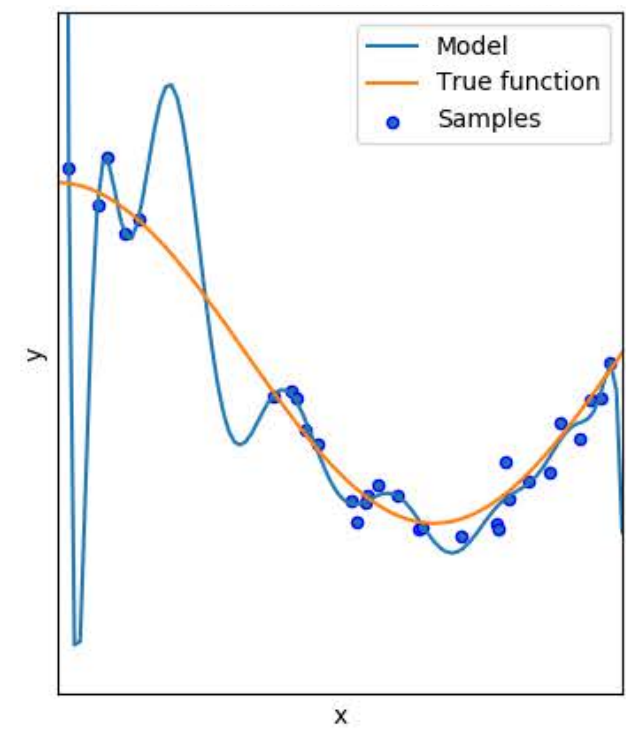
# Linear and Non-Linear Regression



Order 1



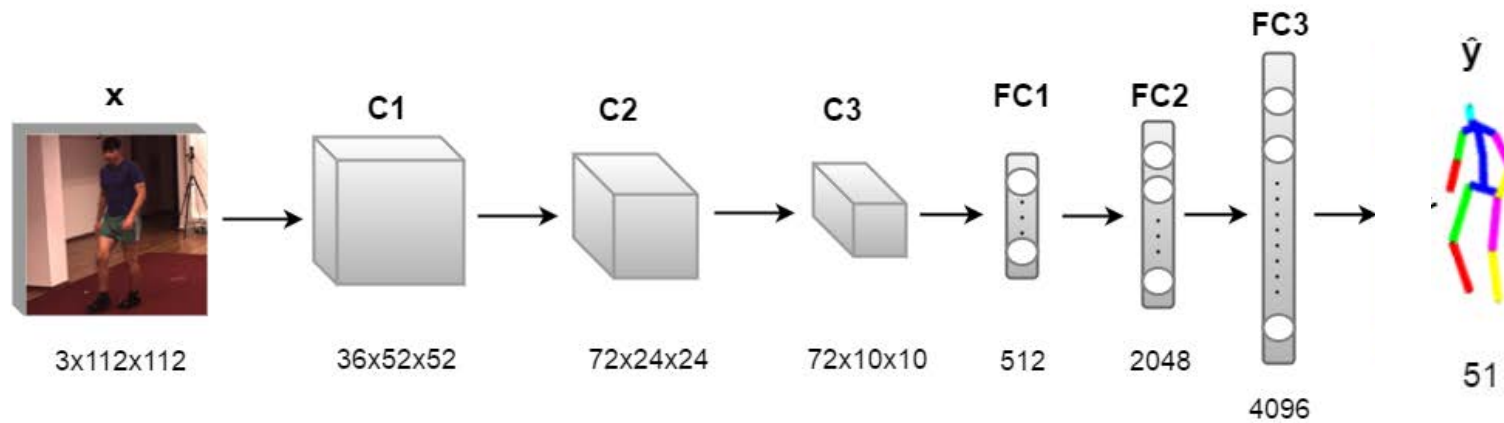
Order 4



Order 15

The trick is to find the best compromise between simplicity and goodness of fit.

# Deep Networks



Input:  $\mathbf{I}$

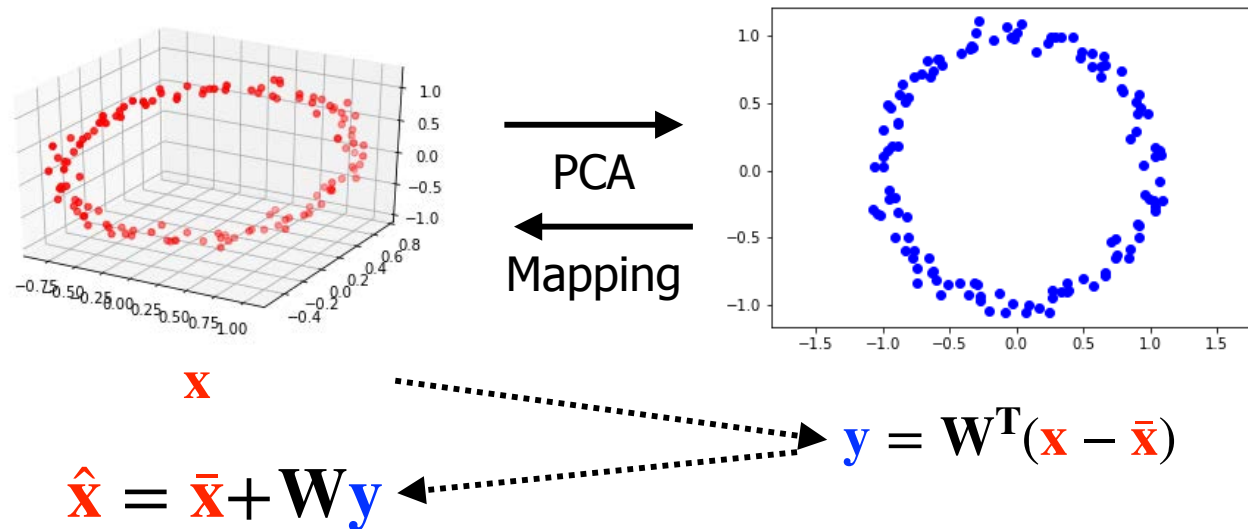
Output:  $\{\mathbf{y}_j\}_{1 \leq j \leq J}$

# Dimensionality Reduction Techniques

- PCA
- LDA
- Autoencoders



# PCA



- This mapping incurs some loss of information.
- However, the corresponding rectangular matrix  $\mathbf{W}$  is the orthogonal matrix that minimizes the reconstruction error

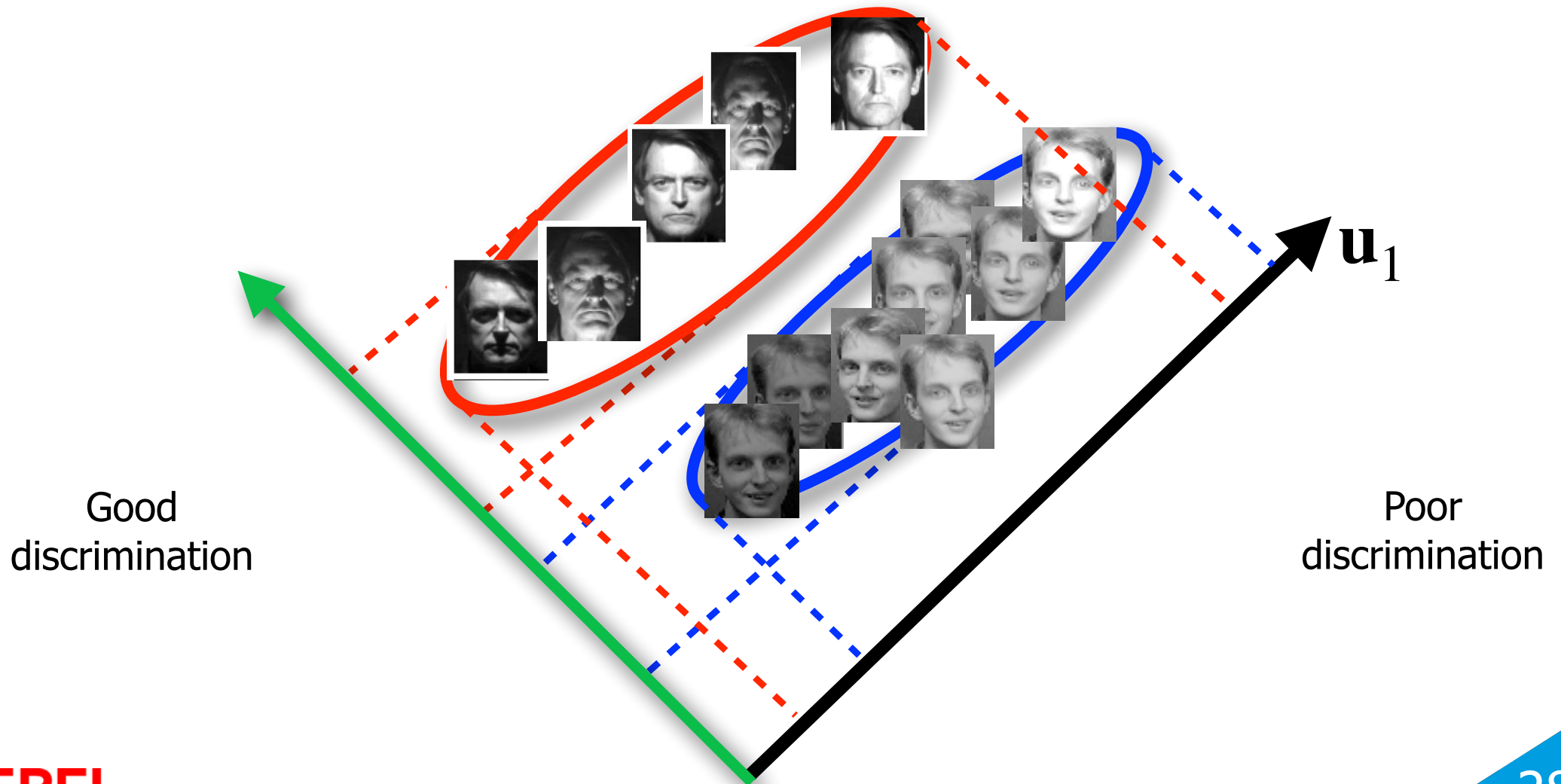
$$e = \|\hat{\mathbf{x}} - \mathbf{x}\|^2$$

where

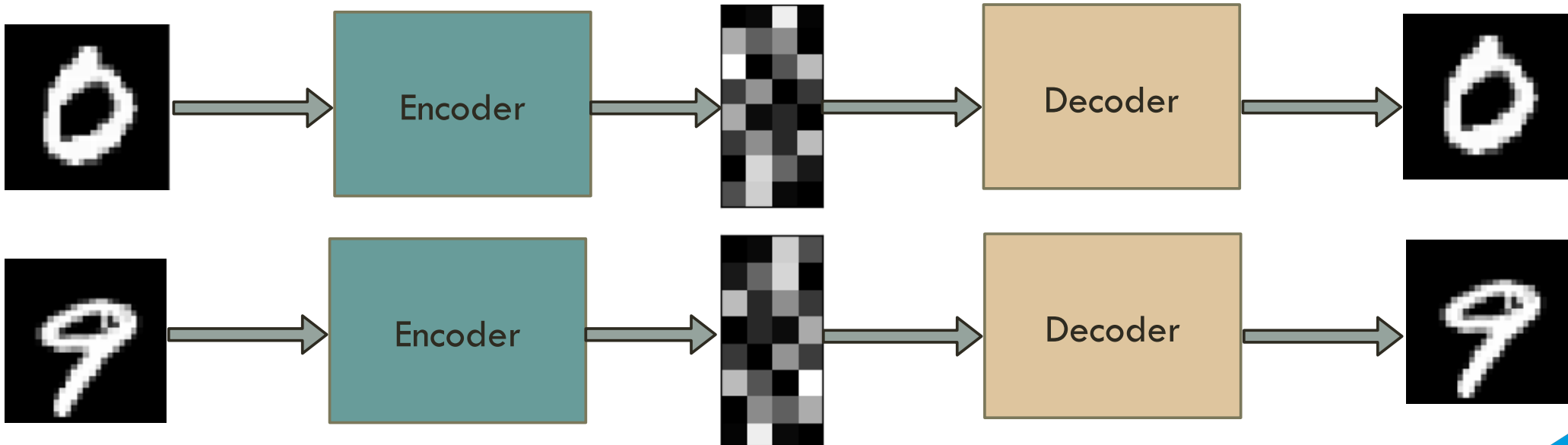
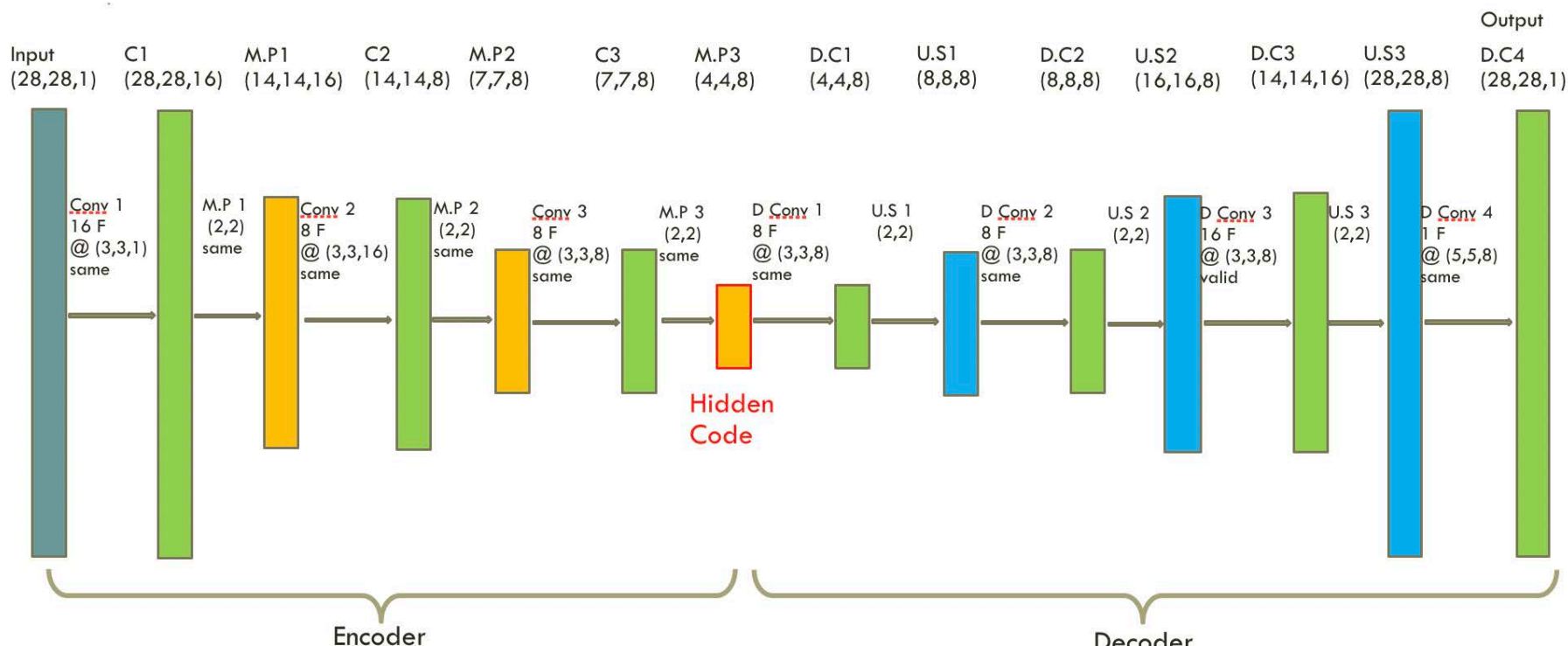
$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{W}\mathbf{y} = \bar{\mathbf{x}} + \mathbf{W}\mathbf{W}^T(\mathbf{x} - \bar{\mathbf{x}})$$

# LDA

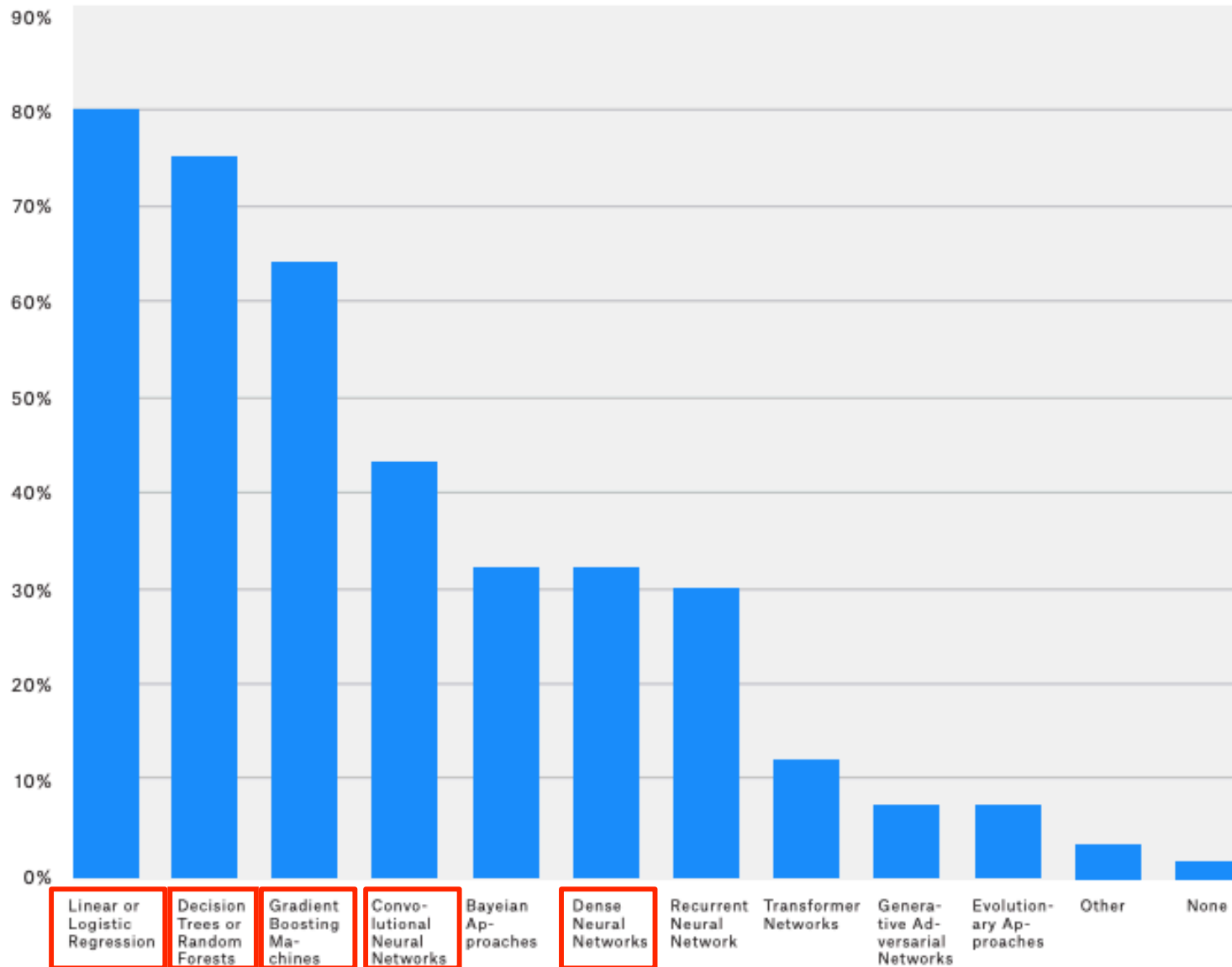
Maximise between class variance and minimize within class variance.



# Autoencoders



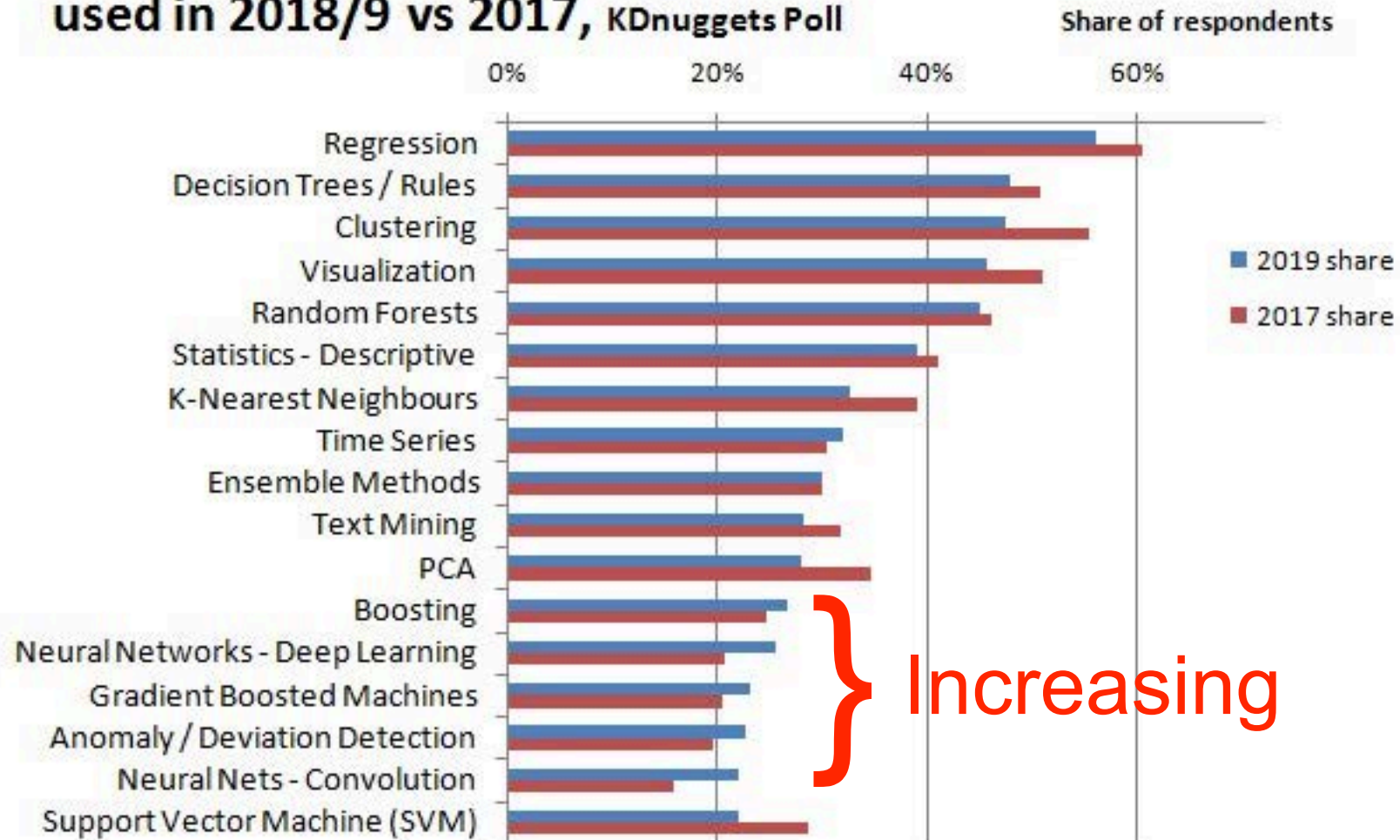
# Kaggle Survey (2019)



What data science methods do you use at work?

# Trends

## Top Data Science, Machine Learning Methods, Algorithms used in 2018/9 vs 2017, KDnuggets Poll



# Logistic Regression on a Massive Scale

## Ad Click Prediction at Google:

Methods such as regularized logistic regression are a natural fit for this problem setting. It is necessary to make predictions many billions of times per day and to quickly update the model as new clicks and non-clicks are observed.

—> The simpler methods are not going away and will probably co-exist with the more sophisticated ones.

# In Conclusion

Rule of thumb:

- Small training set: Use GPs.
- Medium training sets: Use boosted trees.
- Large training set: Use neural nets.

As for all such rules, there are exceptions. Real-time requirements define important ones.

# Exam

- On July 2nd.
- Possibility of extra-mural exam if you **cannot** come.
- 2.0 hours.
- 1 two-sided **hand-written** A4 page of notes.
- Questions on **non-indented** slides on webpage.

See you then.