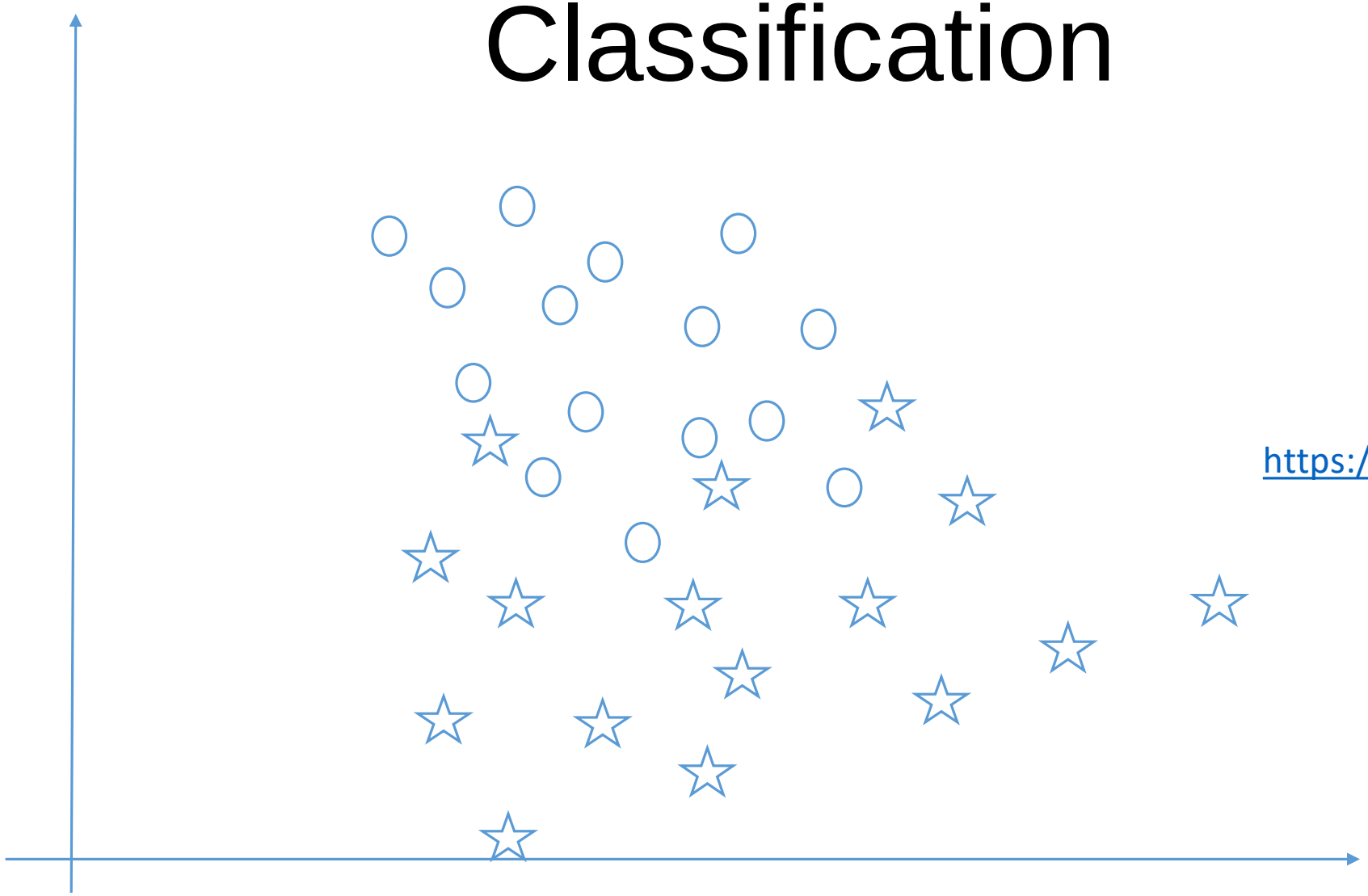
Localization and Other Issues

Sai Ravela

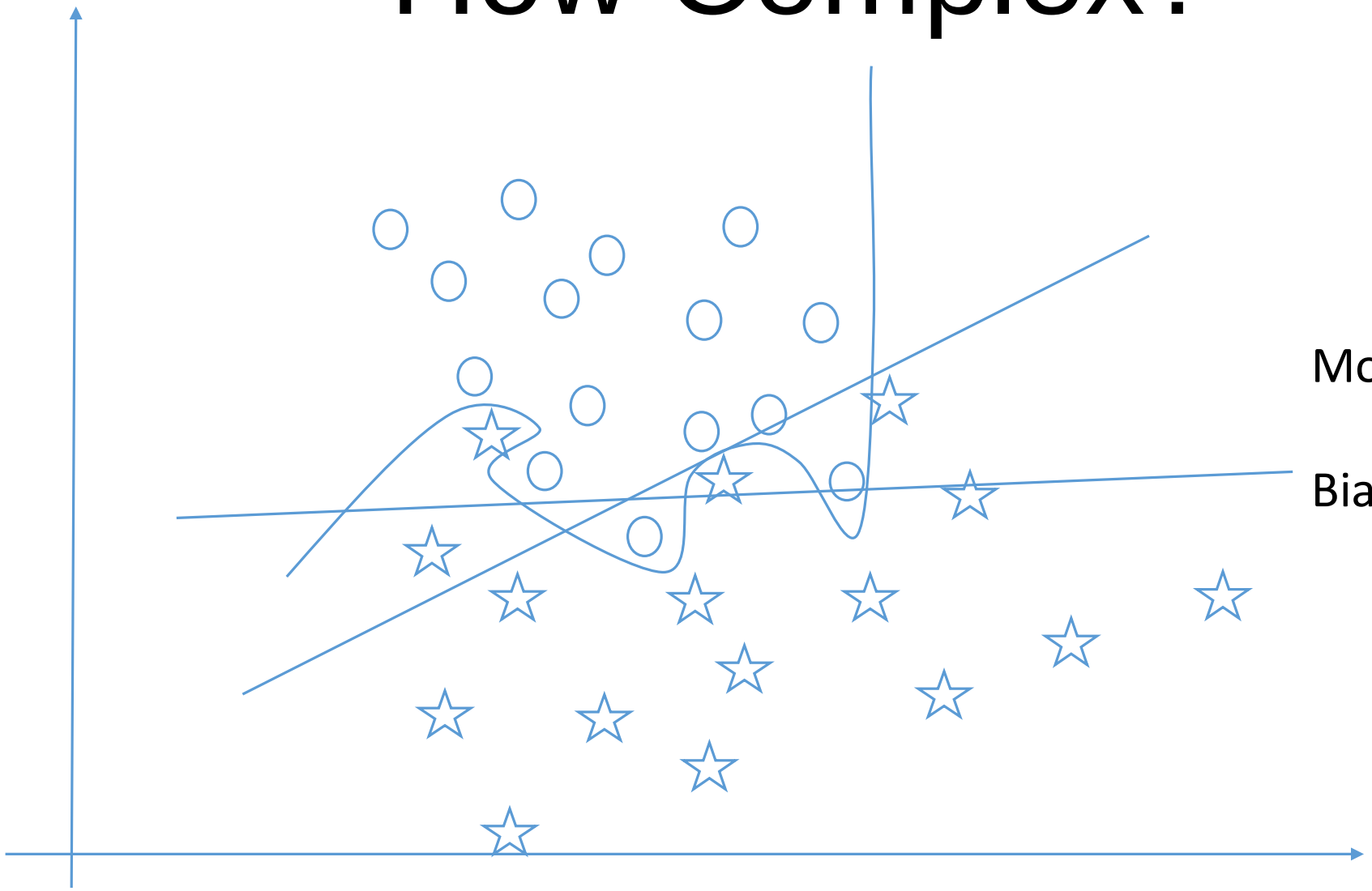
Ex’s and Oh’s and Classification



Loss Functions

https://youtu.be/0uLI6BnVh6w

How Complex?



Model Complexity

Bias-Variance Dilemma

Nearest Neighbors vs Global Regression

- Regression: $y \sim f(\underline{x})$; use all training data to estimate parameters of f, then
  Classification: $c = \begin{cases} 1 & if\ y > 0.5 \\ 0 & otherwise \end{cases}$
- Nearest Neighbor regression: $\mathrm{y} \sim \mathrm{f_x}(\underline{x})$,using training data $\{x_i, c_i \mid x_i \in \mathcal{N}_k(\underline{x})\}$; then classify
- Nearest Neighbor classifier: $\mathrm{c}(\underline{x}) = g(\{c_i | x_i \in \mathcal{N}_k(\underline{x})\})$. The training data is $\{x_i, c_i\}$
- Here $\mathcal{N}_k(\underline{x})$ is the set of k nearest neighbors of $\underline{x}$, whose labels c_i are used by function g to make a decision. For example g could be majority vote

Statistical Model Regression

$$\mathbf{E}\big(y - f(x)\big)^2 = \int\int \big(y - f(x)\big)^2 P(x,y)dxdy$$

$$= \int \left[\int \big(y - f(x)\big)^2 P(y|x)dy\right] P(x)dx$$

$$= \int [\hat{e}\ (x)]^2 P(x)dx$$

$$\arg\min_{\widehat{\boldsymbol{f}_x}} \mathbf{E}_{\mathbf{y}|\mathbf{x}}\big(y - \hat{f}_x\big)^2 = \arg\min_{\widehat{\boldsymbol{f}}} \int \big(y - \hat{f}_x\big)^2 P(y|x)dy$$

$$\Rightarrow \widehat{\boldsymbol{f}}_{\boldsymbol{x}} = \boldsymbol{E}_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y})$$

Take the conditional mean of y at x as the estimate for $f(x)$

For nearest neighbor regression: $\widehat{\boldsymbol{f}}_{\boldsymbol{x}} = \boldsymbol{E}_{\boldsymbol{y}|\mathcal{N}_{\boldsymbol{k}}(\boldsymbol{x})}(\boldsymbol{y})$; using some neighborhood

Statistical Model for Classification

- The variables have $|C|$ categories, say $\mathrm{C} = \{1 \ldots S\}$.
- L is a matrix of dissimilarities, the Loss function, e.g. 0-1 loss function

  $L[c\,, g(\underline{x})] = \begin{cases} 0 & if\ c = g(x) \\ 1 & otherwise \end{cases}$; hit or miss, 1s off diagonal, 0s on it.

  Expected Prediction Error is:

$$\mathbf{E_x}\left[\sum_{s=1}^{S} L[s, g(\underline{x})]\,\mathrm{P}(\mathrm{s}|\underline{x})\right]$$

  For 0-1 loss function, solution for $g(\underline{x})$ is

$$\hat{g}(\underline{x}) = \arg\max_{c} P(c|\underline{x})$$

  Choose the most likely class! ➔ *Bayes Classifier*

Kernels vs. Nearest Neighbors

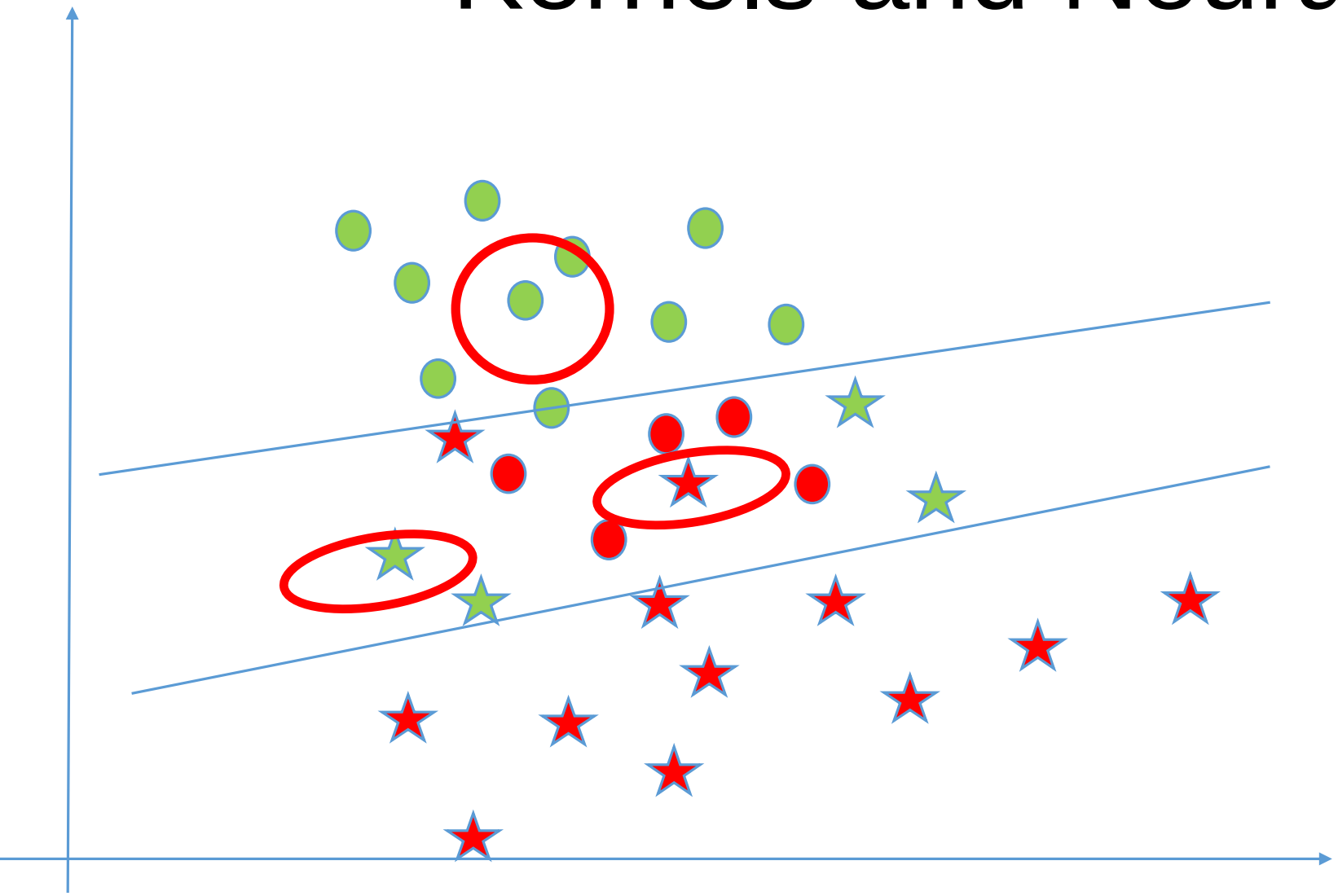
$$y \sim \sum_{i=1}^{N} K_i\left(\underline{x}, \underline{x}_i; \beta_i\right) f\left(\underline{x}_i\right) = \sum_{i=1}^{N} K_i\left(\underline{x}, \underline{x}_i; \beta_i\right) y_i$$

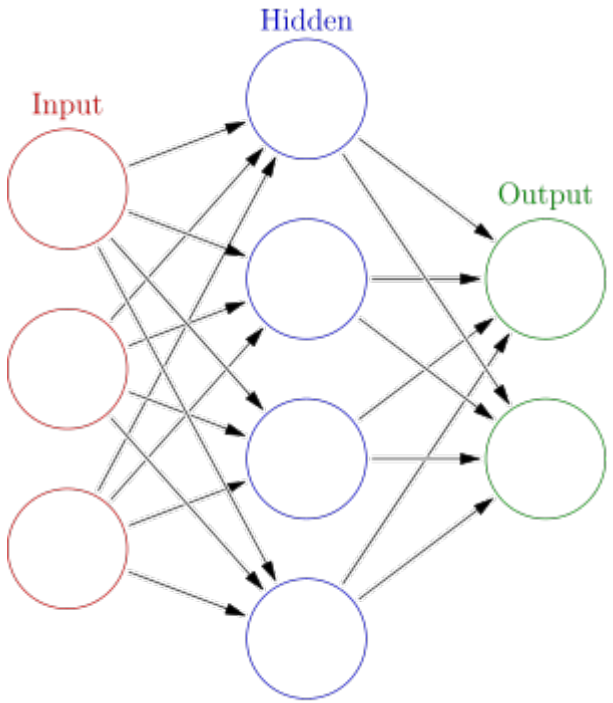
Kernels:

$$K_i\left(\underline{x}, \underline{x}_i\right) = \frac{\mathrm{w_i}}{\sum_j \left(\mathrm{w_j}\right)}, \text{where } \mathrm{w_i} = \frac{1}{\left\|\underline{x} - \underline{x}_i\right\|_2^2 + \epsilon > 0}$$

$$K_i\left(\underline{x}, \underline{x}_i; \sigma_i\right) = \frac{1}{\mathrm{Z}} \mathrm{e}^{-\frac{1}{2}\left(\underline{x} - \underline{x}_i\right)^T \sigma_i^{-2} \left(\underline{x} - \underline{x}_i\right)} \text{ (normalized)}$$

Kernels and Neural Nets





Bases, and other forms of regression

Bases

$$y \sim \sum_{b=1}^{B} f_b(\underline{x})\beta_b$$
$$= \left[\underline{f}^T \underline{\beta}\right](\underline{x})$$

Neural Nets (a single internal layer):

$$y \sim f(\boldsymbol{W}^{\boldsymbol{T}}[\underline{x}\ 1]^T)$$
$$f(\underline{z}) = \sum_i \beta_i \left[\frac{1}{1+\exp(-z_i)}\right]$$

Localization through
Kernels
Basis weights
Regularization
Sparsity
…
Is a key facet of Learning

Regression vs NN

- Regression
  - If a straight line
  - “high bias”
- K-NN
  - Effective degrees of freedom: $\approx \frac{N}{k}$
  - Larger k $\Rightarrow$ more bias, smaller k $\Rightarrow$ more “variance”

- Bias ➔ the parameters change little from one training sample to the next, there is a steady, large error between training data
- Variance ➔ huge changes in parameters, but always fits training data well.

Bias Variance Decomposition

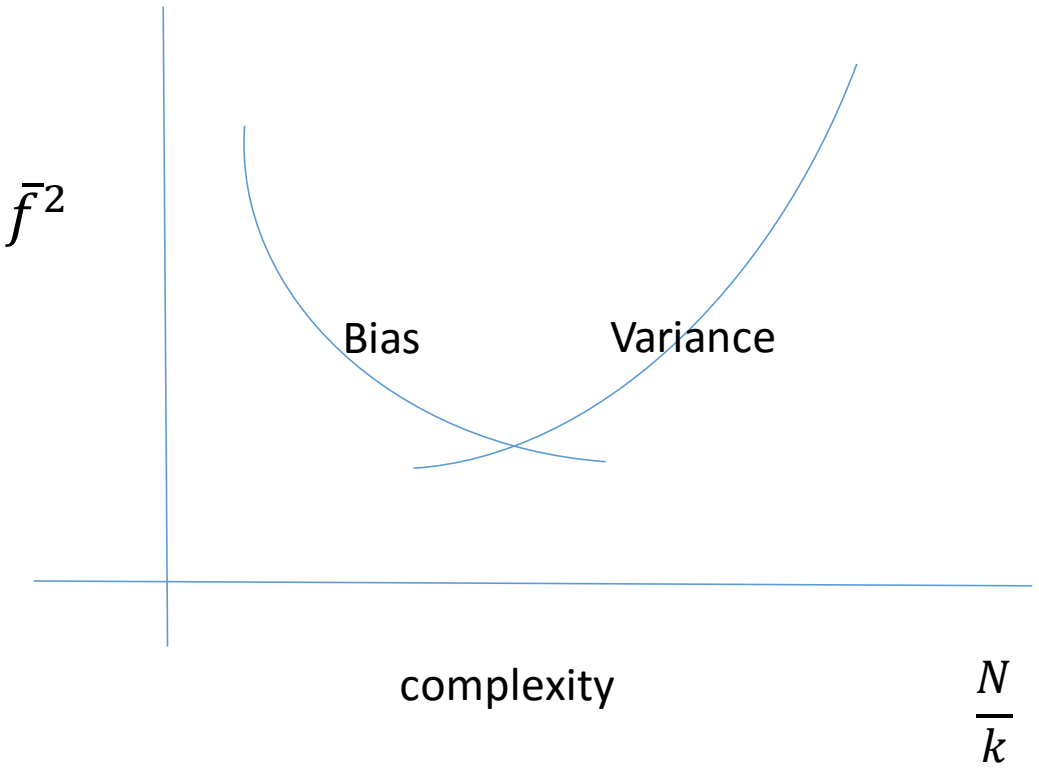
$$y = f(x) + n$$

$$\frac{1}{N}\sum_i \left(y_i - \hat{f}\,(x_i)\right)^2 = \sum_i y_i^2 + \hat{f}_i^{\,2} \quad - 2y_i\hat{f}_i$$

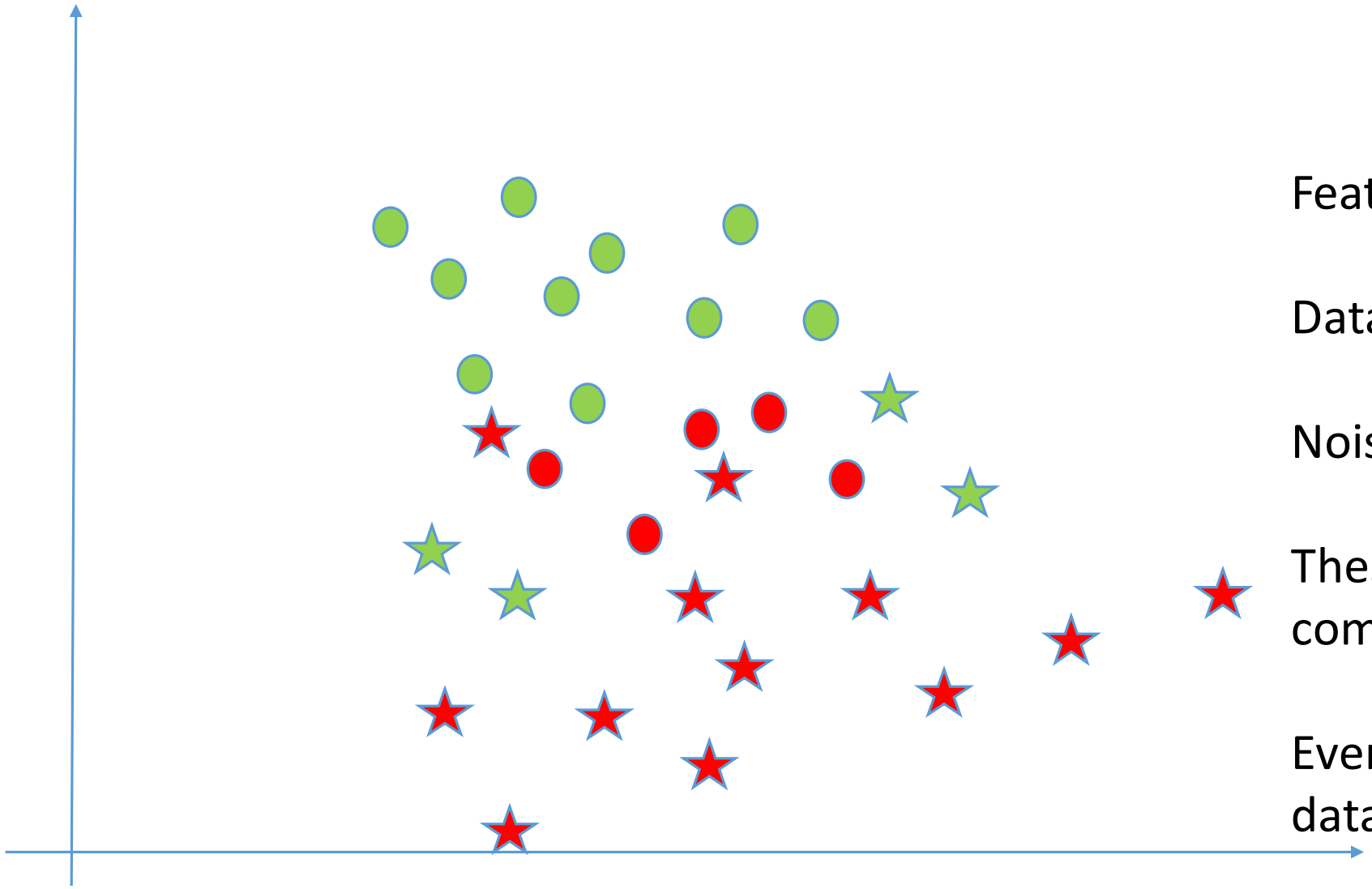
$$= \left[\frac{1}{N}\sum_i y_i^2 - \bar{y}^2\right] + \left[\frac{1}{N}\sum_k \hat{f}_k^2 - \bar{f}^2\right] + \bar{y}^2 - 2\bar{y}\bar{f} + \bar{f}^2$$

$$= \left[\frac{1}{N}\sum_i y_i^2 - \bar{y}^2\right] + \left[\frac{1}{N}\sum_k \hat{f}_k^2 - \bar{f}^2\right] + \left(\bar{y} - \bar{f}\right)^2$$

TV=$noise + variance + bias^2$



Issues: Poor Features, Bad Labels?



Feature Selection

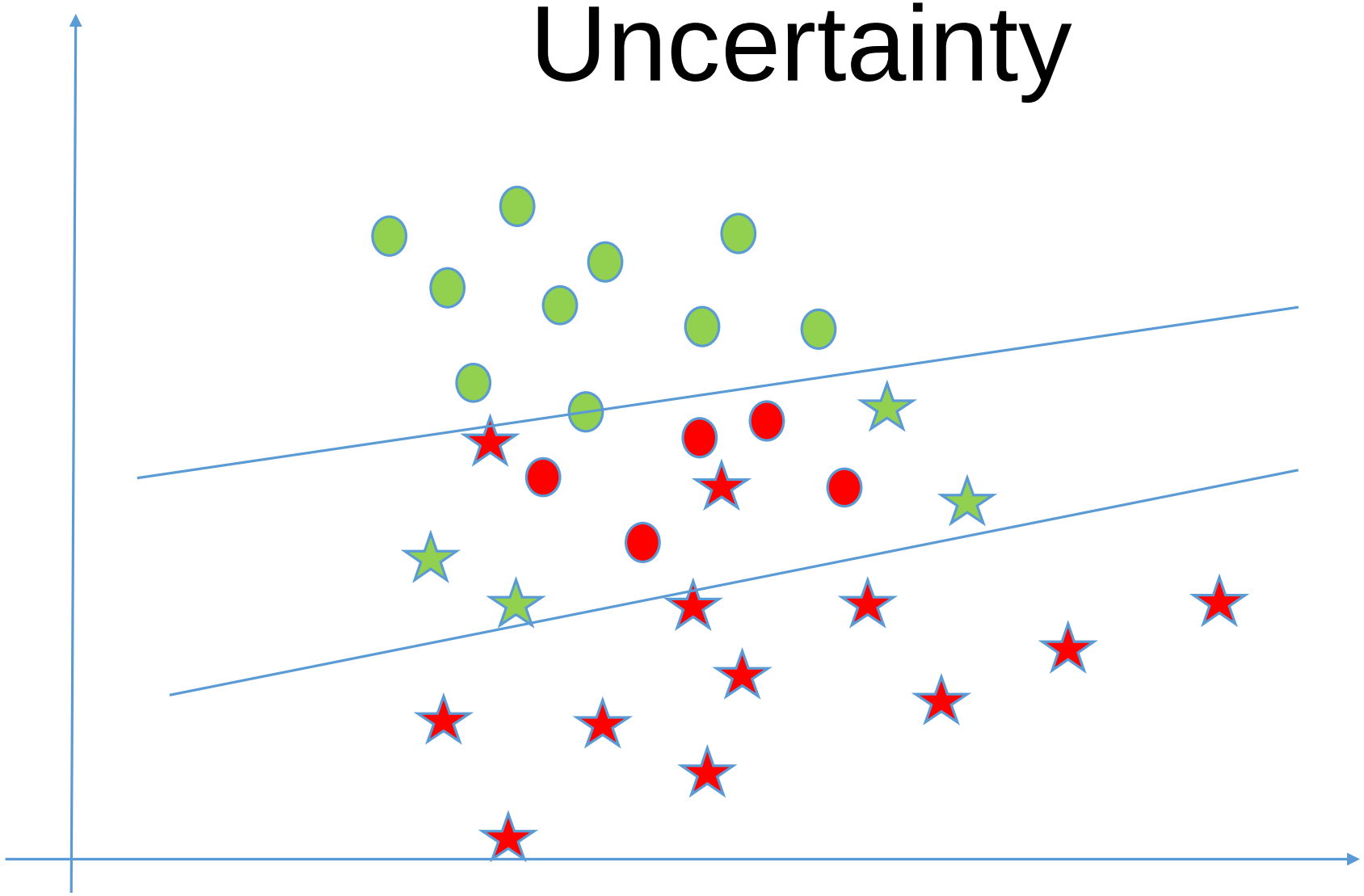
Data Selection

Noisy Labels

These can be common issues

Even generating training data can be difficult

Issues: Model Error and Uncertainty



Uncertainty induced by data labels may imply probabilistic regression and classification is needed

Dimensionality Issues and K-nearest dearest

- Assume data is uniform on a unit hypercube
- For dimension d capturing a fraction r of neighbors requires $r^{\frac{1}{d}}$ edge. As d goes up, this is most of the volume
- If r is reduced, then increased variance.
- In a unit ball, the median distance from origin to closest point of N points is $\left(1-\left(\frac{1}{2}\right)^{\frac{1}{N}}\right)^{\frac{1}{d}}$ so that as d increases, the points are farther apart.
- Finally the sampling density is $N^{\frac{1}{d}}$, so that keeping the same density requires an enormous number of points with increased dimensionality
- ML in high dimensions is a challenge.

Machine Learning

Learning Principles

- Invariance
- Parsimony
- Smoothness
- Sparsity
- Entropy
- Compactness
- Generalization

Learning Knobs

- Loss Functions
- Regularization
- Dimensionality Reduction
- Bias-Variance
- Model Selection
- Feature Selection
- Data Selection

Major Goals

- Prediction
- Characterization
- Explanation

Types of Learning

Supervised

- Training Data {X,Y}
- Task: Y:=f(X)

Unsupervised

- Training Data {X}
- Task: O(X) – Organization of X.

Principles, Tools

- {X,Y} → Learning → M
- $\mathrm{M}(X = x) \Rightarrow Y = \hat{y}$

- {X} → Learning → O
- $\mathrm{O}(X = x) \Rightarrow mode/\text{category}$

Semi-Supervised: Some Data have labels and/or requires internal organization

Modeling by Machine Learning

- Physical Model (Use of physical models to make a prediction)

+ Statistical Model (Use of data to improve prediction)

+ Stochastic Model (Represent the distribution of residuals/joint probabilities)

Machine Learning

The Matrix	Regress	Classify	Estimate	Cluster	Sample Selection	Feature Select	Model Complexity	Loss	Regula rize	Genera lize	Explain
Least Squares	Y	Y	Y	N							
K-NN											
Bayesian MCMC											
Graphical Models											
Trees											
Random Forests											
Ensemble Learning											
Neural Networks											
Manifold Learning											
Mixtures/Kernels											
Regression Machines											
Kernel Machines											
Graph Spectra											