# SC4/SM4 Data Mining and Machine Learning
# Supervised Learning Basics

**Dino Sejdinovic**
Department of Statistics
Oxford

Slides and other materials available at:
http://www.stats.ox.ac.uk/~sejdinov/dmml

# Supervised Learning

**Unsupervised learning**:

- To "extract structure" and postulate hypotheses about data generating process from "unlabelled" observations $x_1, \ldots, x_n$.
- Visualize, summarize and compress data.

**Supervised learning**:

- In addition to the observations of $X$, we have access to their response variables / labels $Y \in \mathcal{Y}$: we observe $\{(x_i, y_i)\}_{i=1}^n$.
- Types of supervised learning:
  - Classification: discrete responses, e.g. $\mathcal{Y} = \{+1, -1\}$ or $\{1, \ldots, K\}$.
  - Regression: a numerical value is observed and $\mathcal{Y} = \mathbb{R}$.

The goal is to accurately predict the response $Y$ on new observations of $X$, i.e., to **learn a function** $f : \mathbb{R}^p \to \mathcal{Y}$, such that $f(X)$ will be close to the true response $Y$.

# Loss function

- Suppose we made a prediction $\hat{Y} = f(X) \in \mathcal{Y}$ based on observation of $X$.
- How good is the prediction? We can use a **loss function** $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ to formalize the quality of the prediction.
- Typical loss functions:
  - **Misclassification loss** (or **0-1 loss**) for classification

$$L(y, f(x)) = \begin{cases} 0 & f(x) = y \\ 1 & f(x) \neq y \end{cases} .$$

  - **Squared loss** for regression

$$L(y, f(x)) = (f(x) - y)^2 .$$

- Many other choices are possible.

# Loss functions for binary classification

Classes are denoted $-1$ and $+1$. Class prediction is $\text{sign}(f(x))$, whereas the magnitude of $f(x)$ represents the "confidence".

- $0/1$ loss $L(y, f(x)) = \mathbb{1}\{yf(x) \leq 0\}$,
  (also called misclassification loss, optimal solution is called the **Bayes classifier** and is given by $f(x) = \text{argmax}_{k \in \{0,1\}} \mathbb{P}(Y = k | X = x)$),
- hinge loss $L(y, f(x)) = (1 - yf(x))_+$
  (used in **support vector machines** - leads to sparse solutions),
- exponential loss $L(y, f(x)) = e^{-yf(x)}$
  (used in **boosting** algorithms - Adaboost),
- logistic loss $L(y, f(x)) = \log\left(1 + e^{-yf(x)}\right)$
  (used in **logistic regression**, associated with a probabilistic model).

The loss can penalize misclassification (wrong sign) as well as the overconfident misclassification (wrong sign and large magnitude) and even underconfident correct classification (correct sign but small magnitude).

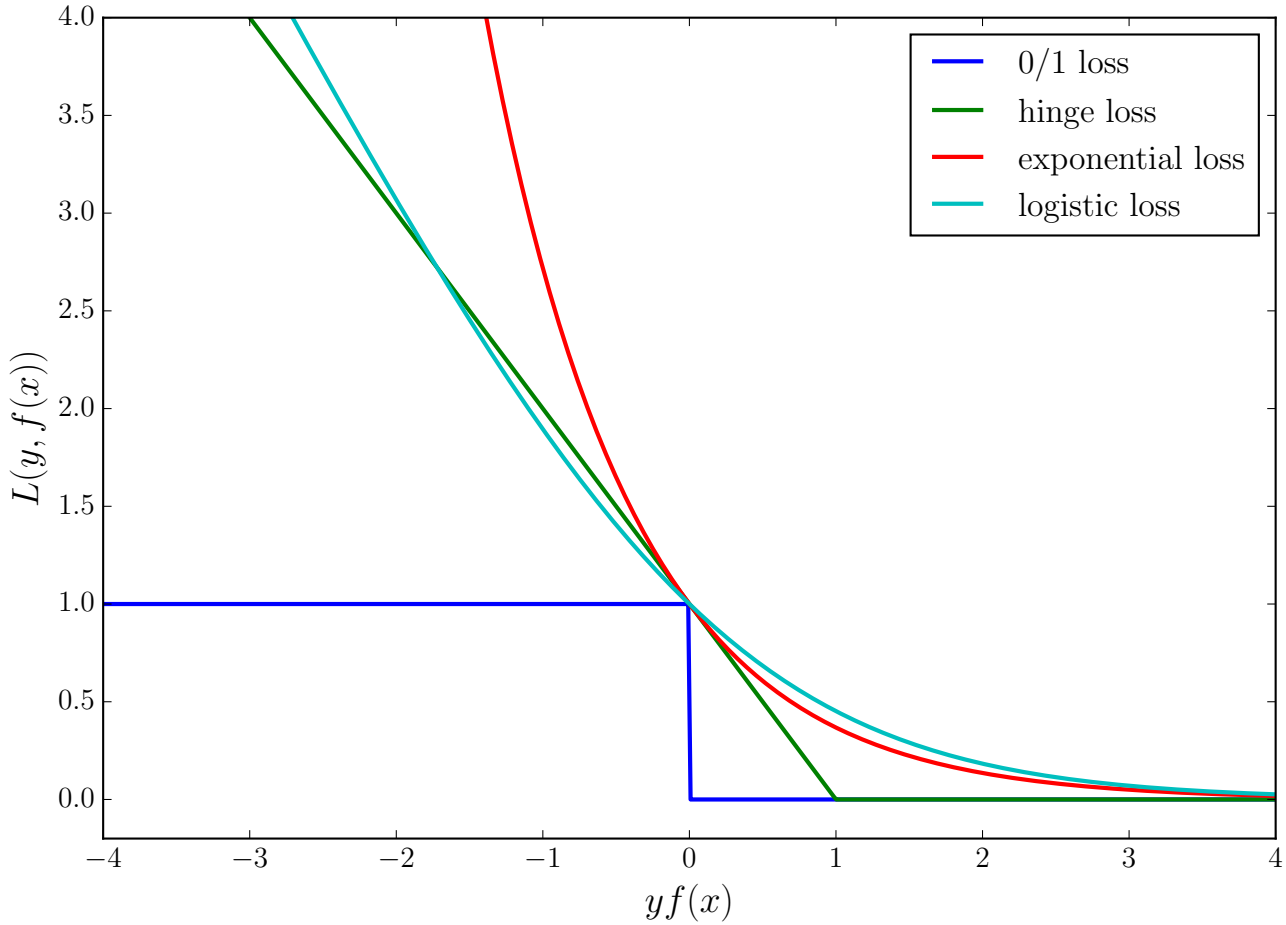



Figure: Loss functions for binary classification

# Loss functions for regression

- squared loss: $L(y,f(x)) = (y - f(x))^2$
  (least squares regression: optimal $f$ is the conditional mean $\mathbb{E}[Y|X = x]$),
- absolute loss: $L(y,f(x)) = |y - f(x)|$
  (less sensitive to outliers: optimal $f$ is the conditional median $\text{med}[Y|X = x]$),
- $\tau$-pinball loss: $L(y,f(x)) = 2\max\{\tau(y - f(x)), (\tau - 1)(y - f(x))\}$ for $\tau \in (0,1)$
  (quantile regression: optimal $f$ is the $\tau$-quantile of $p(y|X = x)$),
- $\epsilon$-insensitive (Vapnik) loss: $L(y,f(x)) = \begin{cases} 0, \text{ if } |y - f(x)| \leq \epsilon, \\ |y - f(x)| - \epsilon, \text{ otherwise.} \end{cases}$
  (**support vector regression** - leads to sparse solutions).

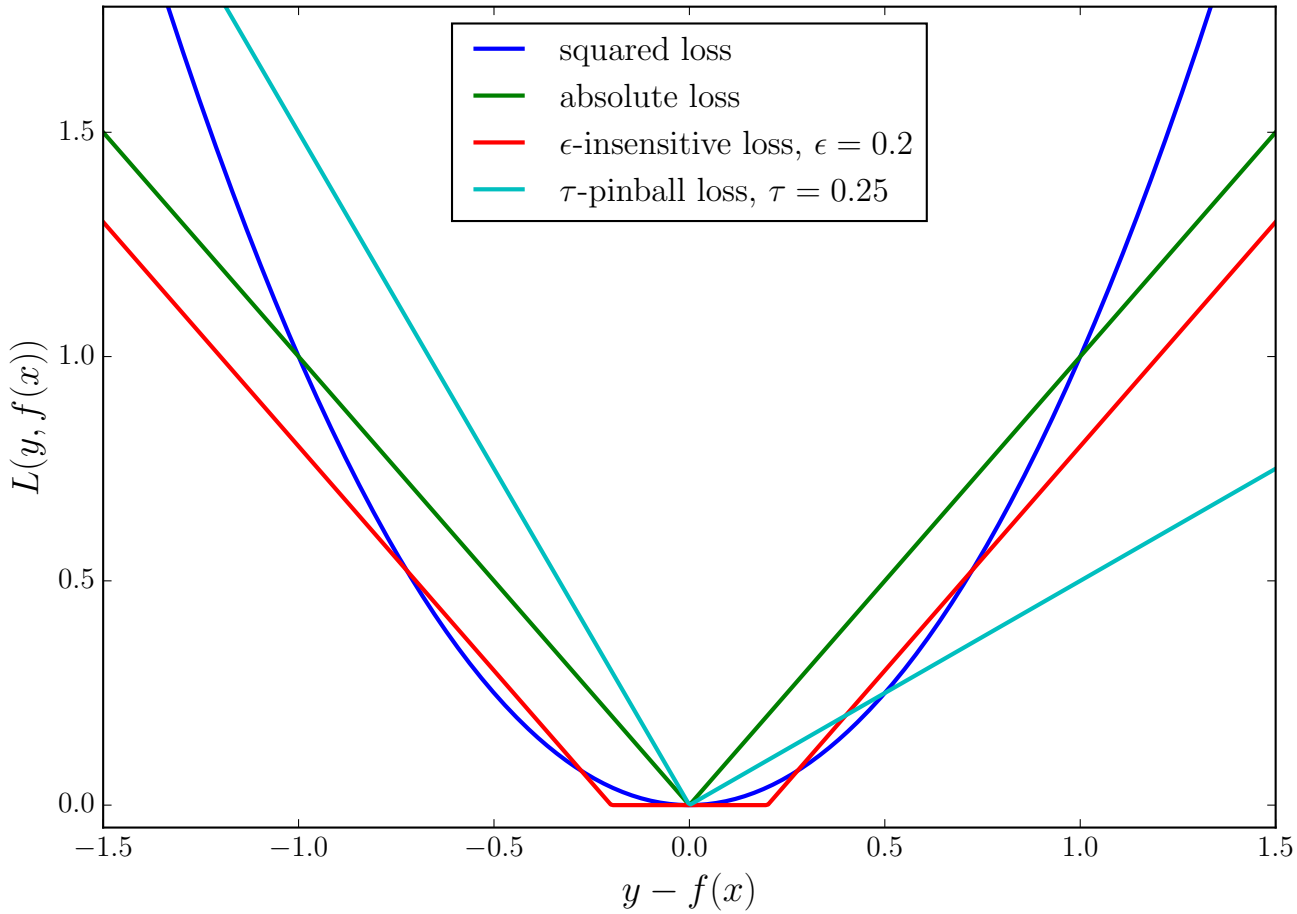



Figure: Loss functions for regression

# Risk

- paired observations $\{(x_i, y_i)\}_{i=1}^n$ viewed as i.i.d. realizations of a random variable $(X, Y)$ on $\mathcal{X} \times \mathcal{Y}$ with joint distribution $P_{XY}$

**Risk**

For a given loss function $L$, the **risk** $R$ of a learned function $f$ is given by the expected loss

$$R(f) = \mathbb{E}_{P_{XY}}\left[L(Y, f(X))\right],$$

where the expectation is with respect to the true (unknown) joint distribution of $(X, Y)$.

- The risk is unknown, but we can compute the **empirical risk**:

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)).$$

# Hypothesis Space and Empirical Risk Minimization

- The goal of learning is to find the function in **hypothesis space** $\mathcal{H}$ which minimises the risk:

$$f_\star = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{X,Y}[L(Y, f(X))]$$

- **Empirical Risk Minimization** (ERM): minimize the empirical risk instead, since we typically do not know $P_{X,Y}$.

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))$$

- Hypothesis space $\mathcal{H}$ is the space of functions $f$ under consideration.
- How complex should we allow functions $f$ to be? If hypothesis space $\mathcal{H}$ is "too large", ERM can lead to **overfitting**.

$$\hat{f}(x) = \begin{cases} y_i & \text{if } x = x_i, \\ 0 & \text{otherwise} \end{cases}$$

will have zero empirical risk, but is useless for generalization, since it has simply "memorized" the dataset.

# Examples of Hypothesis Spaces

Say $\mathcal{X} \subseteq \mathbb{R}^p$.

- all linear functions $f(x) = w^\top x + b$, parametrized by $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$
- consider a specific **nonlinear feature expansion** $\varphi : \mathcal{X} \to \mathbb{R}^D$, with $D > p$ and use functions linear in those features: $f(x) = w^\top \varphi(x) + b$, but nonlinear in the original inputs $\mathcal{X}$, parametrized by $w \in \mathbb{R}^D$ and $b \in \mathbb{R}$. For example, starting with $\mathcal{X} = \mathbb{R}^2$, we can consider $\varphi\left(\begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}\right) = [x_{i1}, x_{i2}, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2]^\top$, such that the resulting function can depend on quadratic and interaction terms as well.
- In the next lecture, we will study an important type of hypothesis space: **Reproducing Kernel Hilbert Space (RKHS)**.

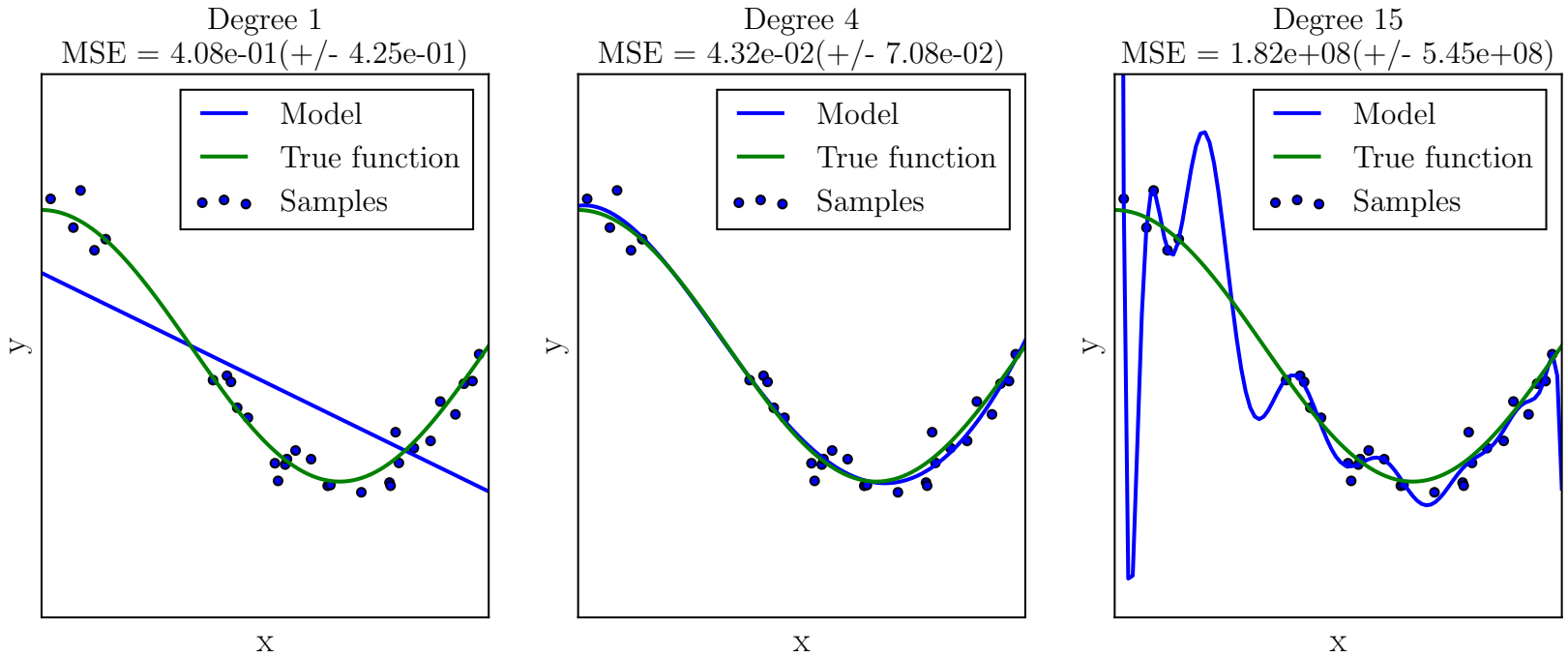



Figure: Underfitting and Overfitting

# Training and Test Performance

- **Training error** is the empirical risk

$$\hat{R}_{\text{tr}}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$$

of the **learned function** $\hat{f}$. For example, for 0-1 loss in classification, this is the number of misclassified training examples **which were used in learning** $\hat{f}$. Note that

$$\mathbb{E}_{P_{XY}} \hat{R}_{\text{tr}}(\hat{f}) \neq R(f).$$

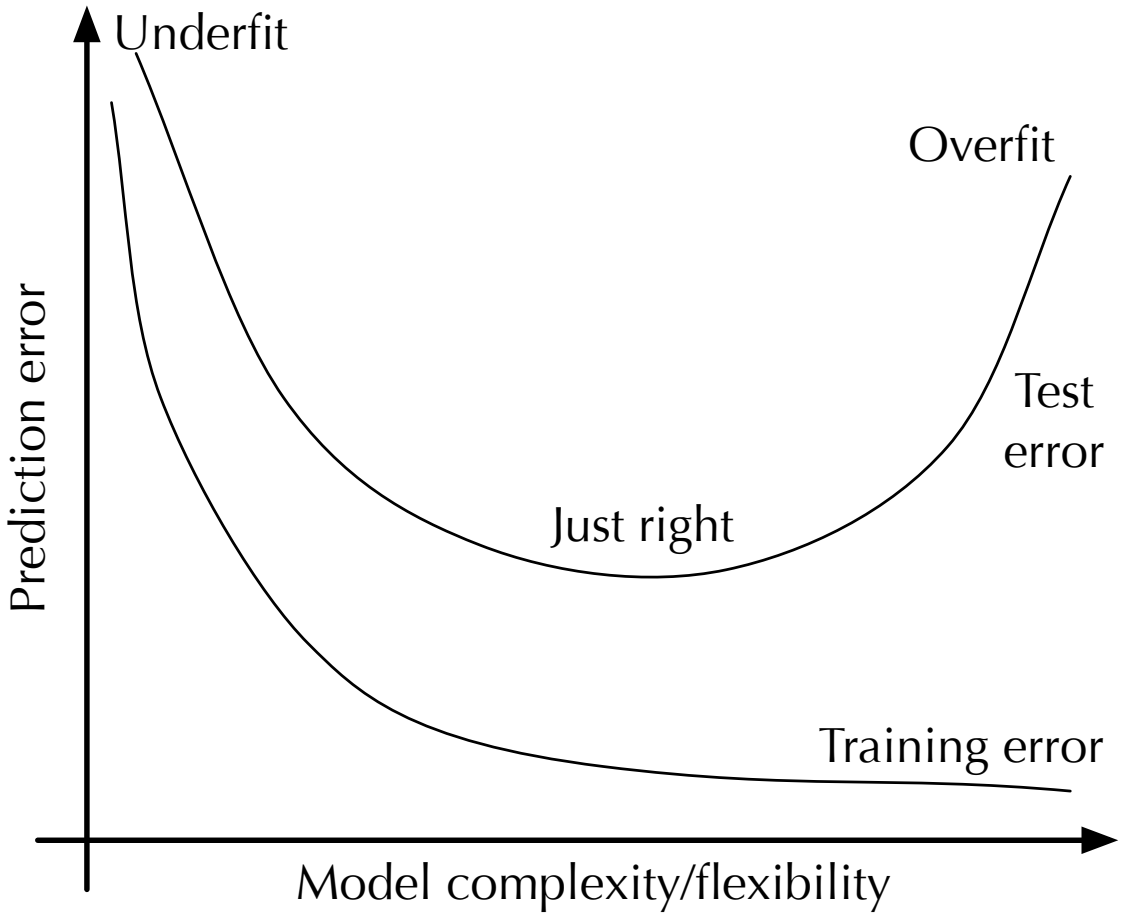
- **Test error** is the empirical risk on **new, previously unseen** observations $\{\tilde{x}_i, \tilde{y}_i\}_{i=1}^{m}$

$$\hat{R}_{\text{tst}}(\hat{f}) = \frac{1}{m}\sum_{i=1}^{m} L(\tilde{y}_i, \hat{f}(\tilde{x}_i))$$

**which were NOT used in learning** $f$.

- Test error tells us how well the learned function **generalizes** to new data ($\mathbb{E}_{P_{XY}} \hat{R}_{\text{tst}}(\hat{f}) \neq R(f)$) and is in general larger than the training error.

# Training and Test Performance




The relationship between training and test error as a function of model complexity.

# regularisation

- Flexible models for high-dimensional problems require many parameters.
- With many parameters, learners can easily overfit.
- **regularisation**: Limit flexibility of model to prevent overfitting.
- Add term **penalizing large values of parameters** $\theta$.

$$\min_\theta \hat{R}(f_\theta) + \lambda \|\theta\|_\rho^\rho = \min_\theta \frac{1}{n} \sum_{i=1}^{n} L(y_i, f_\theta(x_i)) + \lambda \|\theta\|_\rho^\rho$$

where $\rho \geq 1$, and $\|\theta\|_\rho = (\sum_{j=1}^{p} |\theta_j|^\rho)^{1/\rho}$ is the $L_\rho$ norm of $\theta$ (also of interest when $\rho \in [0, 1)$, but is no longer a norm).

- Also known as **shrinkage** methods—parameters are shrunk towards 0.
- $\lambda$ is a **tuning parameter** (or **hyperparameter**) and controls the amount of regularisation, and resulting complexity of the model.

# Types of regularisation

- **Ridge regression** / **Tikhonov regularisation**: $\rho = 2$ (Euclidean norm)
- **LASSO**: $\rho = 1$ (Manhattan norm)
- **Sparsity-inducing** regularisation: $\rho \leq 1$ (nonconvex for $\rho < 1$)
- **Elastic net** regularisation: mixed $L_1/L_2$ penalty:

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} L(y_i, f_\theta(x_i)) + \lambda \left[ (1-\alpha) \|\theta\|_2^2 + \alpha \|\theta\|_1 \right]$$

- directly penalise some notion of **smoothness** of function $f$, e.g. for $\mathcal{X} = \mathbb{R}$, the regularisation term can consist of the **Sobolev norm**

$$\|f\|_{W^1}^2 = \int_{-\infty}^{+\infty} f(x)^2 dx + \int_{-\infty}^{+\infty} f'(x)^2 dx, \tag{1}$$

  which penalises functions with large derivative values.

# $L_1$ promotes sparsity

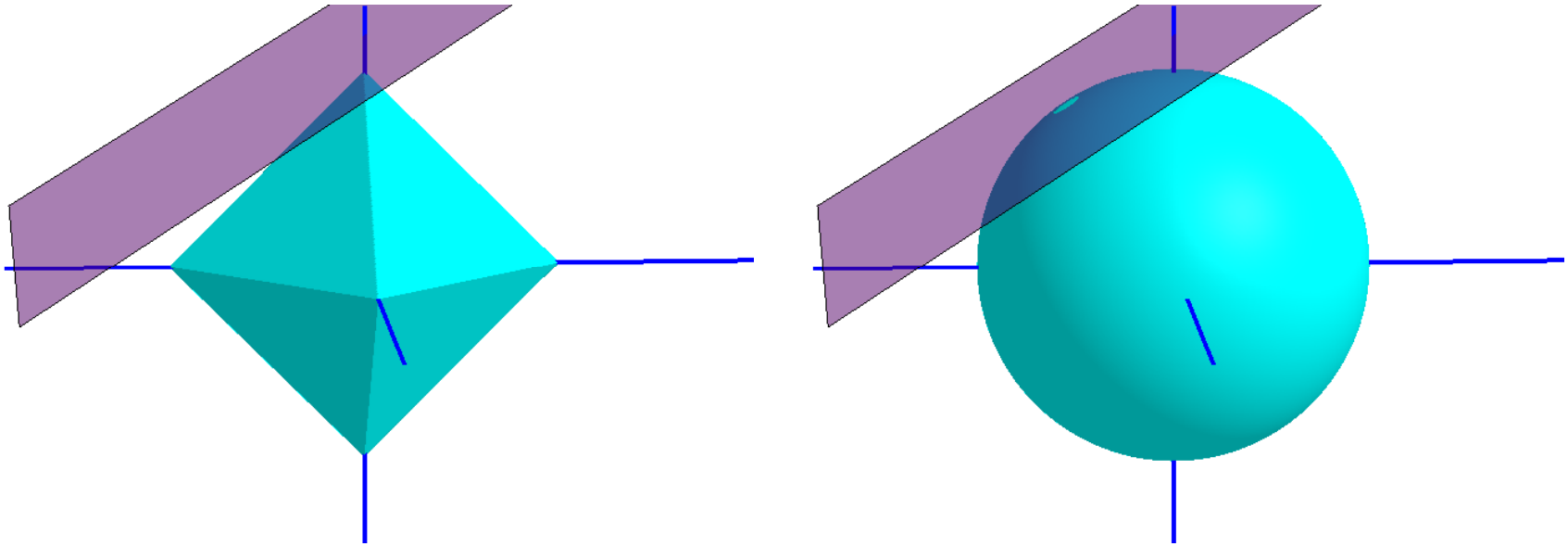


Figure: The intersection between the $L_1$ (left) and the $L_2$ (right) ball with a hyperplane.

$L_1$ regularisation often leads to optimal solutions with many zeros, i.e., the regression function depends only on the (small) number of features with non-zero parameters.

figure from M. Elad, Sparse and Redundant Representations, 2010.