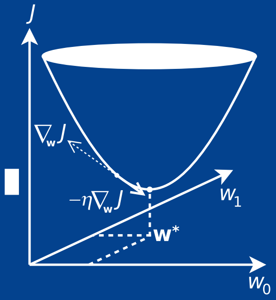
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

MACHINE LEARNING

LESSON 5: Training II

CARSTEN EIE FRIGAARD

SPRING 2019



L05: Training II

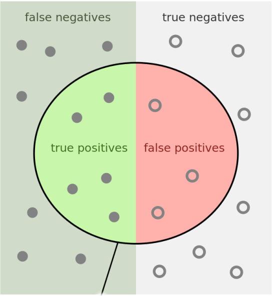
Agenda

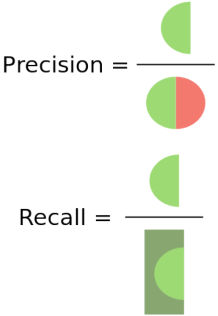
- Important: your feedback (positive as well as negative)...
- Course modification i): *more OPTIONAL* exercices
- Course modification ii): revision to 'Microlearning'
  (only one lecure session, beginning of class)
- Installing `keras` is slow/buggy, see WIKI..
- L04 Training I: Training a linear regression model
  OPTIONAL: Exercise: `L04/linear_regression_1.ipynb`
  OPTIONAL: Exercise: `L04/linear_regression_2.ipynb`
- L05 Training II: **Training and model concepts**
  Exercise: `L05/gradient_descent.ipynb`
  Exercise: `L05/capacity_under_overfitting.ipynb`
  Exercise: `L05/generalization_error.ipynb`
  OPTIONAL: Exercise: `L05/train_test_split.ipynb`

RESUMÉ: Metrics

Precision, recall and accuracy, F_1-score, and confusion matrix

$$
\begin{aligned}
&\text{precision,} && p && = \frac{TP}{TP+FP} \\
&\text{recall (or sensitivity),} && r && = \frac{TP}{TP+FN} \\
&\text{accuracy,} && a && = \frac{TP+TN}{TP+TN+FP+FN} \\
&F_1\text{-score,} && F_1 && = \frac{2pr}{p+r}
\end{aligned}
$$





Confusion Matrix, binary-class data,

$\mathbf{M}_{\text{confusion}} =$

	actual true	actual false
predicted true	TP	FP
predicted false	FN	TN

RESUMÉ: Covariance Matrix

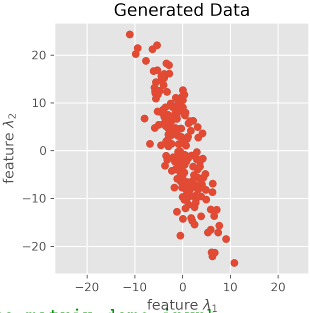
Data matrix for a two-dimensional feature space

$$\mathbf{X} = \begin{array}{c} \begin{matrix} \color{green}{\lambda_1} & \color{green}{\lambda_2} \end{matrix} \\ \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \\ x_1^{(n)} & x_2^{(n)} \end{bmatrix} \end{array} = \begin{bmatrix} 0.35 & -7.62 \\ -4.99 & 13.79 \\ \vdots & \vdots \\ 9.54 & -25.64 \\ 4.21 & -2.25 \end{bmatrix}$$

Covariance matrix, for the two-dimensional feature space

$$\mathbf{\Sigma}(\mathbf{X}) = \begin{bmatrix} \sigma(\lambda_1,\lambda_1) & \sigma(\lambda_1,\lambda_2) \\ \sigma(\lambda_2,\lambda_1) & \sigma(\lambda_2,\lambda_2) \end{bmatrix} = \begin{bmatrix} \sigma(\lambda_1)^2 & \sigma(\lambda_1,\lambda_2) \\ \sigma(\lambda_2,\lambda_1) & \sigma(\lambda_2)^2 \end{bmatrix}$$

with $$\sigma(\lambda_1,\lambda_2) = \frac{1}{n}\sum_{i=1}^{n}(x_1^{(i)} - \mu_{\lambda_1})(x_2^{(i)} - \mu_{\lambda_2})$$



Example: $\mathbf{X}$; a 100 x 2 matrix, see fig..

$$\mathbf{\Sigma}(\mathbf{X}) = \begin{bmatrix} 13.2 & -28.8 \\ -28.8 & 93.3 \end{bmatrix}$$

- $\mathbf{\Sigma}$ is real and symmetric,
- diagonal: the (auto)-variance of a feature, $\sigma(\lambda_j)^2$
- Pearson's r: cross-correlation via cross-covar,
- ~~similar dimension as Confusion matrix~~,
- python implementation: see `L02/Extra/covariance_matrix_demo.ipynb`.

RESUMÉ: Covariance Matrix, Take II

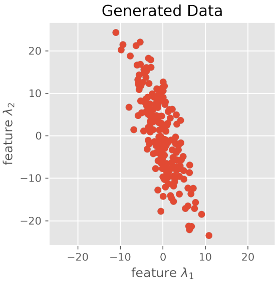
For a dataset, $\mathbf{X}$; features cat and dog; classifying cat/non-cat

Covariance matrix (dim = features x features)

$$\mathbf{\Sigma}(\mathbf{X}) = \begin{bmatrix} \sigma_{\text{cat,cat}} & \sigma_{\text{cat,dog}} \\ \sigma_{\text{dog,cat}} & \sigma_{\text{dog,dog}} \end{bmatrix} = \begin{array}{c|cc} & \text{cat} & \text{dog} \\ \hline \text{cat} & \sigma^2_{\text{cat}} & \sigma_{\text{cat,dog}} \\ \text{dog} & \sigma_{\text{dog,cat}} & \sigma^2_{\text{dog}} \end{array}$$

co-variance

$$\sigma_{\text{cat,dog}} = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)}_{\text{cat}} - \mu_{\text{cat}})(x^{(i)}_{\text{dog}} - \mu_{\text{dog}})$$



Example $\quad \mathbf{\Sigma}(\mathbf{X}) = \begin{bmatrix} 13.2 & -28.8 \\ -28.8 & 93.3 \end{bmatrix}$

Confusion matrix, cat/non-cat

(dim = classes x classes)

$$\mathbf{M} = \begin{bmatrix} \text{TP} & \text{FP} \\ \text{FN} & \text{TN} \end{bmatrix} = \begin{array}{c|cc} & \multicolumn{2}{c}{\textit{actual}} \\ & \text{cat} & \text{non-cat} \\ \hline \text{cat} & \text{T, cat} & \text{F, cat} \\ \text{dog} & \text{F, dog} & \text{T, non-cat} \end{array}$$