COURSE 02457

Signal Processing in Non-linear Systems:

Lecture 4

- The linear model revisited
- Properties of linear discriminants
- The generalization error
- Approximation and regressions
- The bias-variance trade-off
- Test sets
- The learning curve
- Weight decay

Discriminant functions

- A signal detection system divides signal/measurement space in regions $\mathcal{R}$. A set of $discriminant\ functions$ $y_j(\mathbf{x})$ are defined so that

$$y_j(\mathbf{x}) > y_k(\mathbf{x}) \quad j \neq k, \mathbf{x} \in \mathcal{R}_j$$

- Bayes decision theory:

$$y_k(\mathbf{x}) = P(C_k|\mathbf{x})$$

- Special case for binary decisions: A single function defines the decision boundary:

$$y_(\mathbf{x}) = y_1(\mathbf{x}) - y_2(\mathbf{x}) = 0$$

The linear model

- Linear discriminant function for two classes

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

- Terminology: $\mathbf{w}$ are called the weights, and w_{k0} is called the threshold.
- Simplify by dummy input

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

- $\tilde{\mathbf{w}}^T = (w_0, \mathbf{w})$ and $\tilde{\mathbf{x}} = (1, \mathbf{x})$

The linear discriminant

- Linear discriminant functions for multiple classes

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$

- Deciding between two classes j, k

$$y_k(\mathbf{x}) - y_j(\mathbf{x}) = (\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0})$$

- Decision boundary between two classes j, k

$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0$$

Decision regions

- Decision regions of the multiclass linear discriminant are convex (and simply connected)

$$\widehat{\mathbf{x}} = \alpha \mathbf{x}^A + (1-\alpha)\mathbf{x}^B$$

- Let $\mathbf{x}^A, \mathbf{x}^B \in \mathcal{R}_k$, hence $y_k(\mathbf{x}^A) > y_j(\mathbf{x}^A)$ and $y_k(\mathbf{x}^B) > y_j(\mathbf{x}^B)$.

$$\begin{aligned} y_k(\widehat{\mathbf{x}}) &= \mathbf{w}_k^T(\alpha \mathbf{x}^A + (1-\alpha)\mathbf{x}^B) \\ &= \alpha y_k(\mathbf{x}^A) + (1-\alpha) y_k(\mathbf{x}^B) \\ &> \alpha y_j(\mathbf{x}^A) + (1-\alpha) y_j(\mathbf{x}^B) \\ &= \alpha \mathbf{w}_j^T \mathbf{x}^A + (1-\alpha)\mathbf{w}_j^T \mathbf{x}^B \\ &= \mathbf{w}_j^T(\alpha \mathbf{x}^A + (1-\alpha)\mathbf{x}^B) \\ &= y_j(\widehat{\mathbf{x}}) \end{aligned}$$

- Thus all points along the line between $\mathbf{x}^A$ and $\mathbf{x}^B$ are contained in the decision region $\mathcal{R}_k$ (convex and simply connected).

Logistic regression

- Let the class-conditional probability densities for a two-class problem be given by

$$p(\mathbf{x}|C_k) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$

- where the classes have identical covariance matrices $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$
- In this case the posterior probabilities are

$$\begin{aligned} p(C_1|\mathbf{x}) &= \frac{p(\mathbf{x}|C_1)P(C_1)}{p(\mathbf{x}|C_1)P(C_1) + p(\mathbf{x}|C_2)P(C_2)} \\ p(C_2|\mathbf{x}) &= \frac{p(\mathbf{x}|C_2)P(C_2)}{p(\mathbf{x}|C_1)P(C_1) + p(\mathbf{x}|C_2)P(C_2)} \end{aligned}$$

$$\begin{aligned} p(C_1|\mathbf{x}) &= \frac{1}{1 + p(\mathbf{x}|C_2)P(C_2)/p(\mathbf{x}|C_1)P(C_1)} \\ &= \frac{1}{1 + \exp\left(-a(\mathbf{x})\right)} \end{aligned}$$

Logistic regression cont'd

- The logistic regression Bayes decisions are based on

$$p(C_1|\mathbf{x}) \;=\; \frac{1}{1+\exp\left(-a(\mathbf{x})\right)}$$

- $p(C_1|\mathbf{x}) > 0.5$ when the linear discriminant function given by

$$\begin{aligned} a(\mathbf{x}) \;&=\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) \\ &-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \log\frac{P(C_2)}{P(C_1)} \\ &=\; (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 \\ &\quad + \log\frac{P(C_2)}{P(C_1)} \end{aligned}$$

  ... is positive

Logistic regression cont'd

- Hence, we have a recipy for designing a two class detector:
  Estimate the two class mean vectors and the common covariance matrix

$$
\begin{aligned}
\boldsymbol{\mu}_1 &= \frac{1}{N_1}\sum_{n=1}^{N_1}\mathbf{x}^n \\
\boldsymbol{\mu}_2 &= \frac{1}{N_2}\sum_{n=1}^{N_2}\mathbf{x}^n \\
\boldsymbol{\Sigma} &= \frac{1}{N_1+N_2}\left(\sum_{n=1}^{N_1}(\mathbf{x}^n-\boldsymbol{\mu}_1)(\mathbf{x}^n-\boldsymbol{\mu}_1)^T\right. \\
&\quad \left.+\sum_{n=1}^{N_2}(\mathbf{x}^n-\boldsymbol{\mu}_2)(\mathbf{x}^n-\boldsymbol{\mu}_2)^T\right) \\
P(C_1) &= \frac{N_1}{N_1+N_2} \\
P(C_2) &= \frac{N_2}{N_1+N_2}
\end{aligned}
$$

The learning problem

- Supervised learning: Learning relations between sets of variables e.g. between input and output variables, conditional distributions $p(\text{output}|\text{input})$.
- Unsupervised learning: Learning the distribution of a set of variables $p(\text{input})$.

The Bayesian paradigm

- The output density of the measured signals $(t, \mathbf{x})$ is modeled by a parameterized density: $p(t|\mathbf{x}) \sim p(t|\mathbf{x}, \mathbf{w})$.
- Let $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be a *training set*
- Objective: Find the distribution of the parameter vector, $p(\mathbf{w}|\chi)$, hence the parameters are considered stochastic.

The likelihood function

- Let $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be the $training\ set$
- We use Bayes theorem

$$p(\mathbf{w}|\chi) = \frac{p(\chi|\mathbf{w})p(\mathbf{w})}{p(\chi)}$$

- The function $p(\chi|\mathbf{w})$ is called the likelihood function (more correct the likelihood of the parameter vector θ). The density $p(\mathbf{w})$ is called the $a\ priori$ or $prior$ parameter distribution.
- If the prior is "flat" in the neighborhood of the peak of $p(\chi|\mathbf{w})$, we have

$$p(\mathbf{w}|\chi) \propto p(\chi|\mathbf{w})$$

- ...and finding the most probable parameters is equivalent to finding the maximum likelihood parameters.

Maximum likelihood & optimization

- For independent examples, $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$, the likelihood function factorizes

$$p(\chi|\mathbf{w}) = \prod_{n=1}^{N} p(t^n|\mathbf{x}_n, \mathbf{w})p(\mathbf{x}^n) = p(\chi_t|\chi_\mathbf{x}, \mathbf{w}) * p(\chi_\mathbf{x})$$

- Many algorithms are based on minimizing an index or cost function

$$E(\mathbf{w}) = -\log p(\chi_t|\chi_\mathbf{x}, \mathbf{w}) = \sum_{n=1}^{N} -\log p(t^n|\mathbf{x}_n\mathbf{w})$$

Least squares as maximum likelihood

- We seek a conditional density model of the form

$$
\begin{aligned}
y &= f_\theta(\mathbf{x}) + \nu \\
p(y|\mathbf{x}, \sigma^2, \mathbf{w}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - f_{\mathbf{w}}(\mathbf{x}))^2\right) \\
p(\chi_t|\chi_{\mathbf{x}}, \sigma^2, \theta) &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - f_{\mathbf{w}}(\mathbf{x}^n)^2\right) \\
E(\mathbf{w}, \sigma^2) &= \frac{N}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - f_{\mathbf{w}}(\mathbf{x}_n))^2
\end{aligned}
$$

- Hence, maximizing the likelihood for Gaussian noise leads to a least squares problem (for $\mathbf{w}$).
- Note, the noise variance is always given trivially by

$$
\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}(t_n - f_{\widehat{\mathbf{w}}}(\mathbf{x}_n))^2
$$

The generalization error: "The Hidden agenda"

- Let a training set of independent examples be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The *training error pr. example* of the model $p(t|\mathbf{x}, \mathbf{w})$ is given by

$$E = \frac{1}{N} \sum_{n=1}^{N} - \log p(t^n|\mathbf{x}^n, \mathbf{w})$$

  this is what we use to find good parameters $\mathbf{w}$.
- However, what we really want is that the probability of future data points is high, i.e., that the typical cost

$$E^k = - \log p(t^k|\mathbf{x}^k, \mathbf{w})$$

  is low. A model that assigns high probability to all future data point is close to the true model, hence, *a good generalizer*.
- So, let us define *the generalization error*:

$$\begin{aligned} E &= \lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} - \log p(t^k|\mathbf{x}^k, \mathbf{w}) \\ &= \int \int - \log[p(t^k|\mathbf{x}^k, \mathbf{w})] p(t|\mathbf{x}) dt p(\mathbf{x}) d\mathbf{x} \end{aligned}$$

  This is the average (or expected) error on a test datum $(t, \mathbf{x})$.

The generalization error

- Let a training set be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The mean square error of the model $y(\mathbf{x}; \mathbf{w})$ is given by

$$E = \frac{1}{2} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2$$

- Now consider the limit of large sets, the error per example is

$$\begin{aligned} E &= \lim_{N \to \infty} \frac{1}{2N} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \\ &= \frac{1}{2} \int \int (y(\mathbf{x}; \mathbf{w}) - t)^2 p(t, \mathbf{x}) dt d\mathbf{x} \end{aligned}$$

- This is the average (or expected) error on a test datum $(\mathbf{x}, t)$.

The generalization error contd

- The generalization error

$$E = \frac{1}{2} \int \int (y(\mathbf{x}; \mathbf{w}) - t)^2 p(t|\mathbf{x}) p(\mathbf{x}) dt d\mathbf{x}$$

- can be rewritten using the definitions

$$\begin{aligned} < t|\mathbf{x} > &= \int t p(t|\mathbf{x}) dt \\ < t^2|\mathbf{x} > &= \int t^2 p(t|\mathbf{x}) dt \end{aligned}$$

$$\begin{aligned} \{y - t\}^2 &= \{y - < t|\mathbf{x} > + < t|\mathbf{x} > -t\}^2 \\ &= \{y - < t|\mathbf{x} >\}^2 + 2\{y - < t|\mathbf{x} >\}\{< t|\mathbf{x} > -t\} \\ &+ \{< t|\mathbf{x} > -t\}^2 \end{aligned}$$

Regressions

- Then the generalization error becomes

$$\begin{aligned} E &= \frac{1}{2}\int\int (y(\mathbf{x};\mathbf{w}) - t)^2 p(t|\mathbf{x})p(\mathbf{x})dtd\mathbf{x} \\ &= \frac{1}{2}\int\int \{y - < t|\mathbf{x} >\}^2 + 2\{y - < t|\mathbf{x} >\}\{< t|\mathbf{x} > -t\} \\ &+ \{< t|\mathbf{x} > -t\}^2 p(t|\mathbf{x})p(\mathbf{x})dtd\mathbf{x} \end{aligned}$$

- leading to the simplification

$$\begin{aligned} E &= \frac{1}{2}\int (y(\mathbf{x};\mathbf{w}) - < t|\mathbf{x} >)^2 p(\mathbf{x})d\mathbf{x} \\ &+ \frac{1}{2}\int \{< t^2|\mathbf{x} > - < t|\mathbf{x} >^2\} p(\mathbf{x})d\mathbf{x} \end{aligned}$$

- we see that the generalization error is minimal (as function of $y(\mathbf{x};\mathbf{w})$) if

$$y(\mathbf{x};\mathbf{w}) = < t|\mathbf{x} >$$

- The model should output the conditional mean, hence be a "regression"

The bias-variance trade-off

- When training a model we have only a finite training set, hence the model can only find the best approximation minimization the training error

$$E = \frac{1}{2}\sum_{n=1}^{N}(y(\mathbf{x}^n;\mathbf{w}) - t^n)^2$$

- leading to $\mathbf{w}_{\text{optimal}} = \mathbf{w}(\mathcal{D})$, hence the error in a particular test point

$$(y(\mathbf{x};\mathbf{w}(\mathcal{D})) - t)^2$$

  will depend on the particular training set.

- The expected value of this quantity is,

$$\mathcal{E}_{\mathcal{D}}[(y(\mathbf{x};\mathbf{w}(\mathcal{D})) - t)^2]$$

The bias-variance trade-off cont'd

- The expectation

$$\mathcal{E}_{\mathcal{D}}\left[(y(\mathbf{x};\mathbf{w}(\mathcal{D}))- < t|\mathbf{x} >)^2\right]$$

  can be rewritten

$$\begin{aligned}(y(\mathbf{x};\mathbf{w}(\mathcal{D})) \; - \; & < t|\mathbf{x} >)^2 = \\ & \{y(\mathbf{x}) - \mathcal{E}_{\mathcal{D}}[y(\mathbf{x})] + \mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]- < t|\mathbf{x} >\}^2 \\ = \; & \{y(\mathbf{x}) - \mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]\}^2 + \{\mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]- < t|\mathbf{x} >\}^2 \\ + \; & 2\{y(\mathbf{x}) - \mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]\}\{\mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]- < t|\mathbf{x} >\}\end{aligned}$$

- taking expectations w.r.t. $\mathcal{E}_{\mathcal{D}}$,

$$\begin{aligned}\mathcal{E}_{\mathcal{D}}\left[(y(\mathbf{x};\mathbf{w}(\mathcal{D}))- < t|\mathbf{x} >)^2\right] \; = \; & \mathcal{E}_{\mathcal{D}}\left[\{y(\mathbf{x}) - \mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]\}^2\right] \\ + \; & \{\mathcal{E}_{\mathcal{D}}[y(\mathbf{x})]- < t|\mathbf{x} >\}^2\end{aligned}$$

Test sets

- Generalization errors can not be measured, but can be estimated using a finite $test\ set$
- The bias-variance trade-off quantities can be estimated by drawing multiple training sets (can in fact be overlapping i.e. cross-validation)

Crossvalidation

- Generalization errors can not be measured, but can be estimated using a finite $test\ set$
- We would like to use as many examples as possible for training.
- Crossvalidation: Split the data set in V subsets $\mathcal{D}_v$

$$\mathcal{D} = \mathsf{U}_{v=1}^{V} \mathcal{D}_v \tag{1}$$

- For $v = 1, ..., V$ train on $\mathcal{D}/\mathcal{D}_v$ and estimate the test error by

$$\begin{aligned} E_{\text{test}} &= \frac{1}{V} \sum_{v=1}^{V} E_v \\ E_v &= \frac{1}{|\mathcal{D}_v|} \sum_{n \in \mathcal{D}_v} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \end{aligned}$$

- Extreme (and some times optimal): $V = N$, aka leave-one-out crossvalidation

Model capacity and test errors

- The Generalization error depends on the interplay between model flexibility and training set size
- The learning curve is the relation between generalization and training set size: $E_{\text{test}}(N)$ vs. N.
- The generalization error is determined by the complexity of the model and the amount of data N.
- The model complexity is controlled by $regularization$ and by $parameter\ pruning$

Regularization by weight decay

- Weight decay is a means of soft capacity control

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{1}{2}\nu \mathbf{w}^T \mathbf{w}$$

- Analysis of weight decay: Second order Taylor expansion of the costfunction

$$\begin{aligned} E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \sum_j \frac{\partial E}{\partial w_j}(w_j - w_{0,j}) \\ &\quad + \frac{1}{2}\sum_{j,k} \frac{\partial^2 E}{\partial w_j \partial w_k}(w_j - w_{0,j})(w_k - w_{0,k}) \\ E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}_0) \\ &\quad + \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}(\mathbf{w} - \mathbf{w}_0) \end{aligned}$$

- The matrix $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}$ is called the $Hessian$

Weight decay

- Analysis of weight decay:

$$E(\mathbf{w}) \approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}_0) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w} - \mathbf{w}_0)$$

- Hence the minimum solves

$$\frac{\partial E}{\partial \mathbf{w}} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0) = 0$$

$$\mathbf{w}^* = \mathbf{w}_0 - \mathbf{H}^{-1}\frac{\partial E}{\partial \mathbf{w}}$$

Weight decay

- Now if there is a non-zero weight decay

$$\frac{\partial \tilde{E}(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial E}{\partial \mathbf{w}} + \nu \mathbf{w}$$

- Hence the new minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w}^* - \mathbf{w}_0) + \nu(\mathbf{w}^* - \mathbf{w}_0) = 0$$

- while the old minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w}_{\mathrm{old}} - \mathbf{w}_0) = 0$$

- this means that the new and the old minima are related as

$$\mathbf{w}^* - \mathbf{w}_0 = (\mathbf{H}_0 + \nu \mathbf{1})^{-1} \mathbf{H}_0(\mathbf{w}_{\mathrm{old}} - \mathbf{w}_0)$$