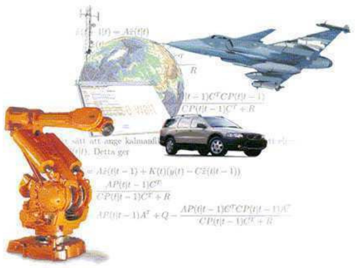
# Welcome to Machine Learning 2011!!



**Thomas Schön**

Division of Automatic Control
Linköping University
Linköping, Sweden.

Email: schon@isy.liu.se,
Phone: 013 - 281373,
Office: House B, Entrance 27.



## What is Machine Learning All About?



*"Machine learning is about learning, reasoning and acting based on data."*



## Outline Lecture 1

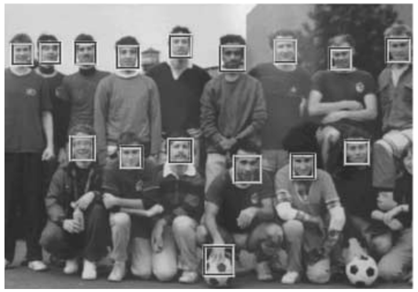


1. Introduction and some motivation
2. Course administration
3. Probability distributions and some basic ideas
   1. Exponential family
   2. Properties of the Multivariate Gaussian
   3. Maximum Likelihood (ML) estimation
   4. Bayesian modelling
   5. Robust statistics ("heavy tails")
   6. Mixture of Gaussians
4. Linear Regression
   1. Linear Basis Function Models
   2. Maximum Likelihood and least squares
   3. Bias variance trade-off
   4. Shrinkage methods (Ridge regression and LASSO)



## Example 1 - Face Detection





Viola, P. and Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features, *Proceedings of the conference on computer vision and pattern recognition (CVPR)*, Kauai, HI, USA, December 2001. Viola, P. A. and

Jones, M. J. (2004) Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137-154.

- Based on boosting (lecture 6)
- Currently implemented in real-time in most cameras, video conferencing equipment, facebook, etc.

## Example 2 - Autonomous Helicopter Aerobatics



Autonomous Helicopter Aerobatics through Apprenticeship Learning, Pieter Abbeel, Adam Coates and Andrew Y. Ng. *International Journal of Robotics Research (IJRR)*, 2011.

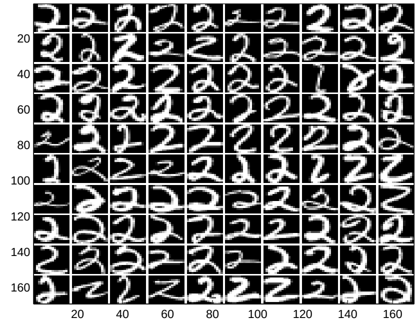
- Learning good controllers for tasks demonstrated by a human. Currently a hot topic in many areas.



## Example 3 - Handwritten Digit Classification



- Input data: $16 \times 16$ grayscale images.
- Task: classify each input image as accurately as possible.
- This data set will be used throughout the course.




Data set available from

`http://www-stat.stanford.edu/~tibs/ElemStatLearn/`



## Example 4 - Animal Detection and Tracking (I/II)



**Volvo förnyar system som ser faror i mörker**

**Volvo utvecklar sitt system för upptäckt av gående som riskerar att bli påkörda till att också fungera i mörker och för stora djur.**

Jag möter Volvos tekniker i den tidiga skymningen i ett hägn med ett 30-tal älgar, dov- och kronhjortar. I en Volvo V70 har det monterats en infraröd kamera framför innerbackspegeln. Kameran är kopplad till en persondator inuti bilen.

Under körning längs vägen i hägnet läser datorn i bilen in de olika djuren från olika vinklar. Både i rörelse och stillastående.

Det är det första fältprovet i en utveckling som ska leda till att bilen själv lär sig känna igen att fara för att viltkollision hotar. Den ska då med ljud- och ljussignaler varna föraren. Om denne inte ingriper är det tänkt att bilen bromsar för att undanröja faran.

**När ska systemet ingripa för djur?**

– Där det finns risk för att förare och passagerare skadas, svarar Andreas Eidehall, som är säkerhetsexpert med djurdetektion som specialitet hos Volvo Personvagnar.

Han förklarar att svårast är att få systemet att inte ingripa i onödan. Därför vägs radarns (monterad i fronten) och den infraröda kamerans information samman.

Andreas Eidehall

– Systemet, som ännu inte har fått något namn, är tänkt att varna förare som är distraherade och ouppmärksamma.

ANDREAS EIDEHALL säger att nackdelen med de nattsynsystem som redan finns på marknaden är att de tvingar föraren att splittra sin uppmärksamhet mellan vindrutan och bildskärmen.

Det blir allt skummare i hägnet i takt med att solen sjunker i väst. Och allt svårare att uppfatta djuren som lockas till vägen med utlagt foder.

Andreas Eidehall berättar att viltolyckor är ett stort problem inte bara i Sverige, utan i många länder där det säljs Volvobilar. I vårt eget land skedde i fjol 40 000 viltolyckor, berättar han. Av dessa olyckor var 13 procent med älg.

**”Systemet, som ännu inte har fått något namn, är tänkt att varna förare som är distraherade och ouppmärksamma.**

Andreas Eidehall, säkerhetsexpert

När klockan har blivit 22.30 är det svårt att urskilja djuren som helt oblygt beträder vägen fastän bilen kommer rullande med påslaget halvljus.

JAG ÅKER MED i bilen och ser på datorns skärm hur väl det infraröda ljuset på gott håll avslöjar djuren i mörkret fastän jag knappt kan skönja dessa när jag tittar ut genom vindrutan.

– Vi ska träna systemet att känna igen djuren, säger Andreas Eidehall som tror att den kommersiella lanseringen i Volvos personbilar ligger några år bort.

JACQUES WALLNER
jacques.wallner@dn.se 08-738 19 61

GUIDE

Bilens dator läser in djuren från olika vinklar.

**Automatskydd mot påkörning**

- **Dagens system** "pedestrian detection with full auto brake" känner igen gående och cyklande i dagsljus, varnar föraren om kollision hotar och ingriper med fullbroms om det behövs för att undvika en olycka.
- **Nästa steg** är att också upptäcka stora vilda djur på och längs vägen
- **Med kombinationen** radar och infraröd kamera upptäcks både människor och stora djur också när det är mörkt.
- **Tanken är att** täcka en 45 grader stor sektor framför bilen, räckvidden bör vara minst 75 meter.



## Example 4 - Animal Detection and Tracking (II/II)



Volvos automatiska skydd mot påkörning är under utveckling. Snart ska det kunna upptäcka stora djur på vägen. FOTO: ANDERS WEJROT

Boosting (lecture 6) promising technology for the detection problem.

## Course Administration



- Lecturer: Thomas Schön, `www.control.isy.liu.se/~schon/`
- This course builds heavily on the Machine Learning course given by Thomas Schön, Umut Orguner and Henrik Ohlsson earlier this year at Linköping University.
- 7 lectures, each 3 hours (Do not cover everything)
- We will try to provide examples of active research throughout the lectures (especially connections to "our" areas)
- Suggested exercises are provided for each lecture
- Written exam, 2 days (48 hours). Code of honour applies as usual
- All course information, including lecture material is available from the course home page

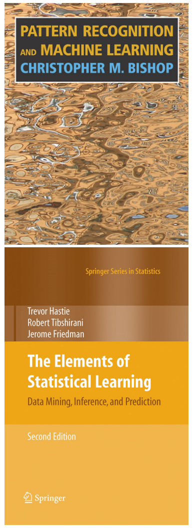
  `www.control.isy.liu.se/~schon/MLLund2011`



## Literature - Course Overview



1. Linear Regression
2. Linear Classification
3. Expectation Maximization (EM)
4. Neural networks
5. Gaussian Processes
6. Support vector machines
7. Clustering
8. Approximate inference
9. Boosting
10. MCMC and sampling methods






## A Few Words About Probability Distributions



- Important in their own right.
- Forms building blocks for more sophisticated probabilistic models.
- Touch upon some important statistical concepts.

See Chapter 2, Appendix B (useful summary) and Wikipedia.



## The Exponential Family



The exponential family of distributions over $x$, parameterized by $\eta$,

$$p(x \mid \eta) = h(x)g(\eta)\exp\left(\eta^T u(x)\right)$$

Some of the members in the exponential family: Bernoulli, Beta, Binomial, Dirichlet, Gamma, Gaussian, Gaussian-Gamma, Gaussian-Wishart, Student's t, Multinomial, Wishart.

## Multivariate Gaussian (I/VI)



$$\mathcal{N}(x;\mu,\Sigma) \triangleq \frac{1}{(2\pi)^{n/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Let us study a partitioned Gaussian,

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

with precision (information) matrix $\Lambda = \Sigma^{-1}$

$$\Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix} = \begin{pmatrix} \Sigma_{aa}^{-1} + \Sigma_{aa}^{-1}\Sigma_{ab}\Delta_a^{-1}\Sigma_{ba}\Sigma_{aa}^{-1} & -\Sigma_{aa}^{-1}\Sigma_{ab}\Delta_a^{-1} \\ -\Delta_a^{-1}\Sigma_{ba}\Sigma_{aa}^{-1} & \Delta_a^{-1} \end{pmatrix}$$

where $\Delta_a = \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}$ is the Schur complement of $\Sigma_{aa}$ in $\Sigma$.



## Multivariate Gaussian (II/VI)



**Theorem (Conditioning)**

*Let $x$ be Gaussian distributed and partitioned $x = \begin{pmatrix} x_a & x_b \end{pmatrix}^T$, then the conditional density $p(x_a \mid x_b)$ is given by*

$$\begin{aligned} p(x_a \mid x_b) &= \mathcal{N}(x_a;\mu_{a|b},\Sigma_{a|b}), \\ \mu_{a|b} &= \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b), \\ \Sigma_{a|b} &= \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}, \end{aligned}$$

*which using the information (precision) matrix can be written,*

$$\begin{aligned} \mu_{a|b} &= \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(x_b - \mu_b), \\ \Sigma_{a|b} &= \Lambda_{aa}^{-1}. \end{aligned}$$



## Multivariate Gaussian (III/VI)



**Theorem (Marginalization)**

*Let $x$ be Gaussian distributed and partitioned $x = \begin{pmatrix} x_a & x_b \end{pmatrix}^T$, then the marginal density $p(x_a)$ is given by*

$$p(x_a) = \mathcal{N}(x_a;\mu_a,\Sigma_{aa}).$$



## Multivariate Gaussian (IV/VI)



**Theorem (Affine transformations)**

*Assume that $x_a$, as well as $x_b$ conditioned on $x_a$, are Gaussian distributed*

$$\begin{aligned} p(x_a) &= \mathcal{N}(x_a;\mu_a,\Sigma_a), \\ p(x_b \mid x_a) &= \mathcal{N}(x_b; Mx_a + b, \Sigma_{b|a}), \end{aligned}$$

*where $M$ is a matrix and $b$ is a constant vector. The marginal density of $x_b$ is then given by*

$$\begin{aligned} p(x_b) &= \mathcal{N}(x_b;\mu_b,\Sigma_b), \\ \mu_b &= M\mu_a + b, \\ \Sigma_b &= \Sigma_{b|a} + M\Sigma_a M^T. \end{aligned}$$

**Theorem (Affine transformations, cont.)**

*The conditional density of $x_a$ given $x_b$ is*

$$p(x_a \mid x_b) = \mathcal{N}(x_a; \mu_{a|b}, \Sigma_{a|b}),$$

*with*

$$\begin{aligned}
\mu_{a|b} &= \Sigma_{a|b} \left( M^T \Sigma_{b|a}^{-1} (x_b - b) + \Sigma_a^{-1} \mu_a \right) \\
&= \mu_a + \Sigma_a M^T \Sigma_b^{-1} (x_b - b - M\mu_a), \\
\Sigma_{a|b} &= \left( \Sigma_a^{-1} + M^T \Sigma_{b|a}^{-1} M \right)^{-1} \\
&= \Sigma_a - \Sigma_a M^T \Sigma_b^{-1} M \Sigma_a.
\end{aligned}$$





Multivariate Gaussian's are important building blocks in more sophisticated models.

For more details, proofs and an example where the Kalman filter is derived using the above theorems is provided,

http://www.control.isy.liu.se/student/graduate/MachineLearning/manipGauss.pdf





The idea underlying maximum likelihood is that the parameters $\theta$ should be chosen in such a way that the measurements $\{x_i\}_{i=1}^N$ are as likely as possible, i.e.,

$$\widehat{\theta} = \arg\max_{\theta} p(x_1, \cdots, x_N \mid \theta).$$

Recall that the likelihood function is not a probability density function over $\theta$ (it is not normalized).





All variables are modelled as random variables.

$$p(\theta \mid x_1, \dots, x_N) \propto p(x_1, \dots, x_N \mid \theta) p(\theta)$$

Posterior $\propto$ Likelihood $\times$ Prior

Provided that it makes sense from a modelling point of view it is convenient to choose prior distributions rendering a computationally tractable posterior distribution.

This leads to the so called *conjugate priors* (if the prior and the posterior have the same functional form, the prior is said to be a conjugate prior for the likelihood).

Again, only make use of conjugate priors if this makes sense from a modelling point of view!

## Conjugate Priors - Example 1 (I/II)



Let $X = \{x_n\}_{n=1}^N$ be independent identically distributed (iid) observations of $x \sim \mathcal{N}(\mu, \sigma^2)$. Assume that the variance $\sigma^2$ is known.

The likelihood is given by

$$p(X \mid \mu) = \prod_{n=1}^{N} p(x_n \mid \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2\right) \tag{1}$$

If we choose the prior as $p(\mu) = \mathcal{N}(\mu \mid \mu_0, \sigma_0^2)$, the posterior will also be Gaussian. Hence, this prior is a conjugate prior for the likelihood (1).



## Conjugate Priors - Example 1 (II/II)



The resulting posterior is

$$p(\mu \mid X) = \mathcal{N}(\mu_B, \sigma_B^2),$$

where the parameters are given by

$$\mu_B = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML},$$
$$\frac{1}{\sigma_B^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

The ML estimate of the mean is

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n.$$



## Conjugate Priors – Some Examples



| **Likelihood** | **Model Parameters** | **Conjugate Prior** |
|---|---|---|
| Normal (known mean) | Variance | Inverse-Gamma |
| Multivariate Normal (known mean) | Precision | Wishart |
| Multivariate Normal (known mean) | Covariance | Inverse-Wishart |
| Multivariate Normal | Mean and covariance | Normal-Inverse-Wishart |
| Multivariate Normal | Mean and precision | Normal-Wishart |
| Exponential | Rate | Gamma |



## Conjugate Prior is Just One of Many Possibilities!



Note that using a conjugate prior is *just one* of the many possible choices for modelling the prior! If it makes sense, use it, since it leads to simple calculations.

Let's have a look at an example where we do not make use of the conjugate prior and end up in a useful and interesting result.

Linear regression models the relationship between a continuous target variable $t$ and an (input) variable $x$ according to

$$t_i = w_0 + w_1 x_{1,i} + w_2 x_{2,i} + \cdots + w_D x_{D,i} + \epsilon_i$$
$$= w^T \phi(x_i) + \epsilon_i,$$

where $\phi(x_i) = \begin{pmatrix} 1 & x_{1,i} & \dots & x_{D,i} \end{pmatrix}^T$ and $i = 1, \dots, N$.

## Conjugate Prior is Just One of Many Possibilities!



Let $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$, resulting in the following likelihood

$$p(t_i \mid w) = \mathcal{N}(t_i \mid w^T \phi(x_i), \sigma^2).$$

Let us now assume the $w_i$ to be independent and Laplacian distributed (i.e. not conjugate prior), $w_i \sim \mathcal{L}(0, 2\sigma^2/\lambda)$

Def. (Laplacian distribution) $\mathcal{L}(x \mid a, b) = \frac{1}{2b} \exp\left(-\frac{|x-a|}{b}\right)$.

The resulting MAP estimate is given by,

$$\boxed{w^{\text{MAP}} = \arg\max_{w} \sum_{i=1}^{N} (t_i - w^T \phi(x_i))^2 + \lambda \sum_{i=1}^{D} |w_i|}$$

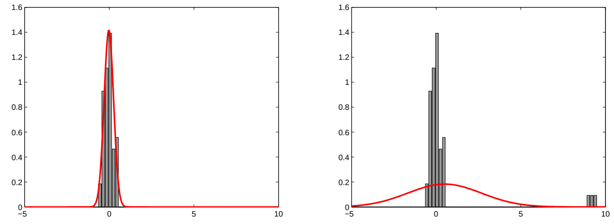
Known as the LASSO and it leads to sparse estimates.



## Robust Statistics



Modelling the error as a Gaussian leads to very high sensitivity to outliers in the data. This is due to the fact that the Gaussian assigns very low probability to points far from the mean. The Gaussian is said to have *"thin tails"*.



Two possible solutions

1. Model using a distribution with "heavy tails".
2. Outlier detection models



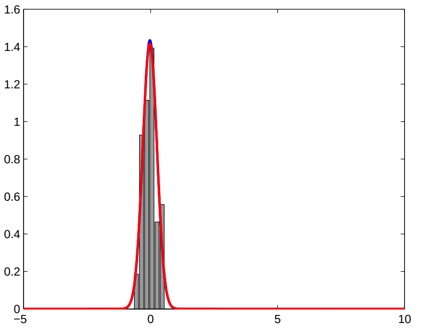
## Example: Heavy Tails (I/III)



Generate $N = 50$ samples,

$$x \sim \mathcal{N}(0, 0.1)$$

Plot showing a realization (gray histogram) and the corresponding ML estimate of a Gaussian (red) and a Student's t-distribution (blue).




Note that (as expected?) the red curve sits on top of the blue curve.



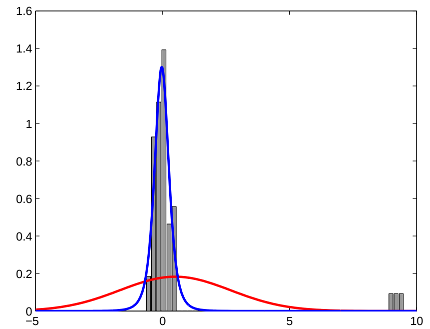
## Example: Heavy Tails (II/III)



Let us now add $3$ outliers $9, 9.2$ and $9.5$ to the data set.

Plot showing resulting ML estimate of a Gaussian (red) and a Student's t-distribution (blue).

Clearly the Student's t-distribution is a better model here!

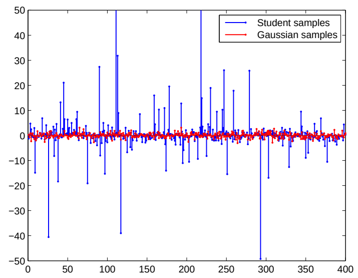


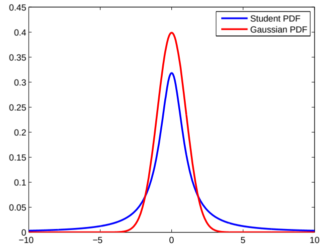
## Example: Heavy Tails (III/III)

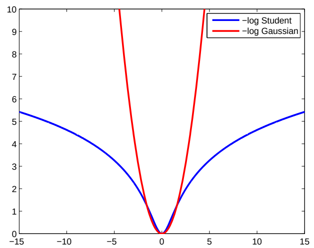


Below: 400 samples from a Student's t-distribution and a Gaussian distribution.

Right: The corresponding pdf's and negative log-likelihoods.






## Outlier Detection Models



Model the data as if it comes from a mixture of two Gaussians,

$$\begin{aligned} p(x_i) &= p(x_i \mid k_i = 0)p(k_i = 0) + p(x_i \mid k_i = 1)p(k_i = 1) \\ &= \mathcal{N}(0, \sigma^2)p(k_i = 0) + \mathcal{N}(0, \alpha\sigma^2)p(k_i = 1). \end{aligned}$$

where $\alpha > 1$, $p(k_i = 0)$ is the probability that the sample is OK and $p(k_i = 1)$ is the probability that the sample is an outlier.

Note the similarity between these two "robustifications". The Student's t-distribution is an infinite mixture of Gaussians, where the mixing is controlled by the $\nu$-parameter. The outlier detection model above consists of a sum of two Gaussians.



## Summary – Robust Statistics



- Do not use distributions with thin tails (non-robust) if there are outliers present. Use more realistic robust "heavy tailed" distribution such as the Student's t-distribution or simply a mixture of two Gaussians.
- A nice account on robustness is available in Section 3.1 in Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A. (2000) Bundle Adjustment - A Modern Synthesis. In: *Vision algorithms: theory and practice*. Lecture Notes in Computer Science, Vol 1883. Springer, Berlin, pp 152-177. `http://dx.doi.org/10.1007/3-540-44480-7_21`



## Important Message!



*Given the computational tools that we have today it can be rewarding to resist the Gaussian convenience!!*

We will try to repeat and illustrate this message throughout the course using theory and examples.

## One Recent Example


Student's t-distribution (this lecture) + variational Bayes (lecture 8) for estimation of AR-models published in this month's issue of IEEE TSP,

Christmas, J. and Everson, R. Robust Autoregression: Student-t Innovations Using Variational Bayes. *IEEE Transactions on Signal Processing*, 59(1): 48 - 57, Jan. 2011.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5582315



## Outline Lecture 1


4. Linear Regression
   1. Linear Basis Function Models
   2. Maximum Likelihood and least squares
   3. Bias variance trade-off
   4. Shrinkage methods (Ridge regression and LASSO)



## Commonly used Basis Functions


In using nonlinear basis functions, $y(x, w)$ can be a nonlinear function in the input variable $x$ (still linear in $w$).

- Global (in the sense that a small change in $x$ affects all basis functions) basis function
  1. Polynomial (see illustrative example in Section 1.1) (ex. identity $\phi(x) = x$)
- Local (in the sense that a small change in $x$ only affects the nearby basis functions) basis function
  1. Gaussian
  2. Sigmoidal



## Linear Regression Model on Matrix Form


It is commonly convenient to write the linear regression model

$$t_n = w^T \phi(x_n) + \epsilon_n, \qquad n = 1, \dots, N,$$

where $w = \begin{pmatrix} w_0 & w_1 & \dots & w_{M-1} \end{pmatrix}^T$ and $\phi = \begin{pmatrix} 1 & \phi_1(x_n) & \dots & \phi_{M-1}(x_n) \end{pmatrix}^T$ on matrix form

$$T = \Phi w + E,$$

where

$$T = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix} \quad \Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{pmatrix} \quad E = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix}$$

## Maximum Likelihood and Least Squares (I/IV)



In our linear regression model,

$$t_n = w^T\phi(x_n) + \epsilon_n,$$

assume that $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$ (i.i.d.). This results in the following likelihood function

$$p(t_n \mid w, \beta) = \mathcal{N}(w^T\phi(x_n), \beta^{-1})$$

Note that this is a slight abuse of notation, $p_{w,\beta}(t_n)$ or $p(t_n; w, \beta)$ would have been better, since $w$ and $\beta$ are both considered deterministic parameters in ML.



## Maximum Likelihood and Least Squares (II/IV)



The available training data consisting of $N$ input variables $X = \{x_i\}_{i=1}^N$ and the corresponding target variables $T = \{t_i\}_{i=1}^N$.

According to our assumption on the noise, the likelihood function is give by

$$p(T \mid w, \beta) = \prod_{n=1}^{N} p(t_n \mid w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid w^T\phi(x_n), \beta^{-1})$$

which results in the log-likelihood function

$$L(w, \beta) \triangleq \ln p(t_1, \dots, t_n \mid w, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n \mid w^T\phi(x_n), \beta^{-1})$$
$$= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2$$



## Maximum Likelihood and Least Squares (III/IV)



The maximum likelihood problem now amounts to solving

$$\underset{w,\beta}{\arg\max}\ L(w, \beta)$$

Setting the derivative $\frac{\partial L}{\partial w} = \beta\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2\phi(x_n)^T$ equal to $0$ gives the following ML estimate for $w$

$$\widehat{w}^{\text{ML}} = \underbrace{(\Phi^T\Phi)^{-1}\Phi^T}_{\Phi^\dagger} T,$$

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{pmatrix}$$

Note that if $\Phi^T\Phi$ is singular (or close to) we can fix this by adding $\lambda I$, i.e.,

$$\widehat{w}^{\text{RR}} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T T,$$



## Maximum Likelihood and Least Squares (IV/IV)



Maximizing the log-likelihood function $L(w, \beta)$ w.r.t. $\beta$ results in the following estimate for $\beta$

$$\frac{1}{\widehat{\beta}^{\text{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \widehat{w}^{\text{ML}}\phi(x_n)\right)^2$$

Finally, note that if we are only interested in $w$, the log-likelihood function is proportional to

$$\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2,$$

which clearly shows that assuming a Gaussian noise model and making use of Maximum Likelihood (ML) corresponds to a Least Squares (LS) problem.

## Interpretation of the Gauss-Markov Theorem



The least squares estimator has the smallest mean square error (MSE) of all linear estimators with no bias, BUT there may exist a biased estimator with lower MSE.

Two classes of potentially biased estimators

1. Subset selection methods
2. Shrinkage methods

This is intimately connected to the bias-variance trade-off

- We will give a system identification example related to ridge regression to illustrate the bias-variance trade-off.
- See Section 3.2 for a slightly more abstract (but very informative) account of the bias-variance trade-off. (this is a perfect topic for discussions during the exercise sessions!)



## Interpretation of RR Using the SVD of $\Phi$



By studying the SVD of $\Phi$ it can be shown that ridge regression projects the measurements onto the principal components of $\Phi$ and then shrinks the coefficients of low-variance components more than the coefficients of high-variance components.

(See Section 3.4.1. in HTF for details.)



## Bias-Variance Tradeoff - Example (I/IV)



(Ex. 2.3 in Henrik Ohlsson's PhD thesis) Consider a SISO system

$$y_t = \sum_{k=1}^{n} g_k^0 u_{t-k} + e_t, \tag{2}$$

where $u_t$ denotes the input, $y_t$ denotes the output, $e_t$ denotes white noise ($\mathbf{E}\{e\} = 0$ and $\mathbf{E}\{e_t e_s\} = \sigma^2 \delta(t-s)$) and $\{g_k^0\}_{k=1}^n$ denote the impulse response of the system.

Recall that the *impulse response* is the output $y_t$ when $u_t = \delta(t)$ is used in (2), which results in

$$y_t = \begin{cases} g_t^0 + e_t & t = 1, \dots, n, \\ e_t & t > n. \end{cases}$$



## Bias-Variance Tradeoff - Example (II/IV)



The task is now to estimate the impulse response using an $n$th order FIR model,

$$y_t = w^T \phi_t + e_t,$$

where

$$\phi_t = \begin{pmatrix} u_{t-1} & \dots & u_{t-n} \end{pmatrix}^T, \qquad w \in \mathbb{R}^n$$

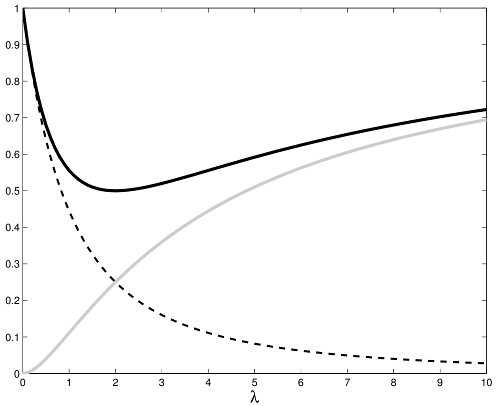
Let us use Ridge regression (RR),

$$\widehat{w}^{RR} = \arg\min_{w} \|Y - \Phi w\|_2^2 + \lambda w^T w.$$

to find the parameters $w$.

## Bias-Variance Tradeoff - Example (III/IV)






Squared bias (gray line)

$$\left(\mathbf{E}_{\widehat{w}}\left(\widehat{w}^T\phi_*\right) - w_0^T\phi_*\right)^2$$

Variance (dashed line)

$$\mathbf{E}_{\widehat{w}}\left(\left(\mathbf{E}_{\widehat{w}}\left(\widehat{w}^T\phi_*\right) - \widehat{w}^T\phi_*\right)^2\right)$$

MSE (black line)

MSE = (bias)$^2$ + variance + $\sigma^2$



## Bias-Variance Tradeoff - Example (IV/IV)



Flexible models will have a low bias and high variance and more "restricted" models will have high bias and low variance.

The model with the best predictive capabilities is the one which strikes the best tradeoff between bias and variance.

Recent contributions on impulse response identification using regularization, see

Pillonetto, G. and De Nicolao, G. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81-93, January 2010.

Chen, T., Ohlsson, H and Ljung, L. On the Estimation of Transfer Functions, Regularizations and Gaussian Processes - Revisited. *In Proceedings of the 18th IFAC World Congress*, Milan, Italy, September 2011. (accepted for publication)



## Lasso



The Lasso was introduced during lecture $1$ as the MAP estimate when a Laplacian prior is assigned to the parameters. Alternatively we can motivate the Lasso as the solution to

$$\begin{aligned} \min_{w} \quad & \textstyle\sum_{n=1}^{N}\left(t_n - w^T\phi(x_n)\right)^2 \\ \text{s.t.} \quad & \textstyle\sum_{j=0}^{M-1}|w_j| \leq \eta \end{aligned}$$

which using a Lagrange multiplier $\lambda$ can be stated

$$\min_{w} \sum_{n=1}^{N}\left(t_n - w^T\phi(x_n)\right)^2 + \lambda\sum_{j=0}^{M-1}|w_j|$$

The difference to ridge regression is simply that Lasso make use of the $\ell_1$-norm $\sum_{j=0}^{M-1}|w_j|$, rather than the $\ell_2$-norm $\sum_{j=0}^{M-1}w_j^2$ used in ridge regression in shrinking the parameters.

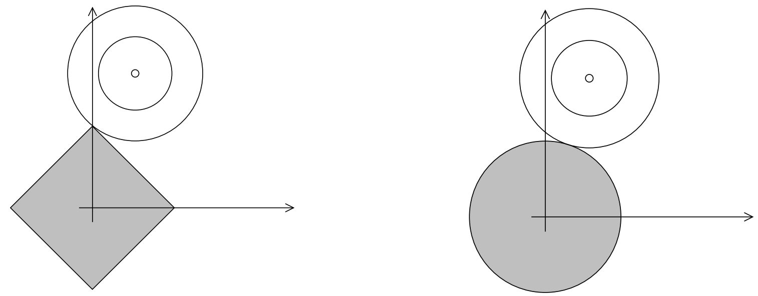


## Graphical Illustration of Lasso and RR



Lasso

Ridge Regression (RR)



The circles are contours of the least squares cost function (LS estimate in the middle). The constraint regions are shown in gray $|w_0| + |w_1| \leq \eta$ (Lasso) and $w_0^2 + w_1^2 \leq \eta$ (RR). The shape of the constraints motivates why Lasso often leads to sparseness.

## Implementing Lasso

The $\ell_1$-regularized least squares problem (lasso)

$$\min_{w} \|T - \Phi w\|_2^2 + \lambda \|w\|_1 \tag{3}$$

YALMIP code solving (3). Download: http://users.isy.liu.se/johanl/yalmip/

```
w=sdpvar(M,1);
ops=sdpsettings('verbose',0);
solvesdp([],(T-Phi*w)'*(T-Phi*w) + lambda*norm(w,1),ops)
```

CVX code solving (3). Download: http://cvxr.com/cvx/

```
cvx_begin
variable w(M)
minimize((T-Phi*w)'*(y-Phi*w) + lambda*norm(w,1))
cvx_end
```

A MATLAB package dedicated to $\ell_1$-regularized least squares problems is `l1_ls`. Download: http://www.stanford.edu/~boyd/l1_ls/





## A Few Concepts to Summarize Lecture 1

**Supervised learning:** The data consists of both input and output signals (e.g., regressions and classification).

**Unsupervised learning:** The data consists of output signals only (e.g., clustering).

**Reinforcement learning:** Finding suitable actions (control signals) in a given situation in order to maximize a reward. (Very similar to control theory)

**Conjugate prior:** If the posterior distribution is in the same family as the prior distribution, the prior and posterior are *conjugate distributions* and the prior is called a conjugate prior for the likelihood.

**Maximum likelihood:** Choose the parameters such that the observations are as likely as possible.

**Linear regression:** Models the relationship between a continuous target variable $t$ and a possibly nonlinear function $\phi(x)$ of the input variables.

**Maximum a Posteriori (MAP):** A point estimate obtained by maximizing the posterior distribution. Corresponds to a mode of the posterior distribution.

**Ridge regression:** An $\ell_2$-regularized least squares problem used to solve the linear regression problem resulting in potentially biased estimates. A.k.a. Tikhonov regularization.

**Lasso:** An $\ell_1$-regularized least squares problem used to solve the linear regression problem resulting in potentially biased estimates. The Lasso typically produce sparse estimates.