# Probabilistic Models in Machine Learning

Gunnar Rätsch[1], Cheng Soon Ong[1,2], Petra Philips[1]

[1] Friedrich Miescher Laboratory, Tübingen
[2] MPI for Biological Cybernetics, Tübingen

Vorlesung WS 2007/2008
Eberhard-Karls-Universität Tübingen

20 November 2007

http://www.fml.mpg.de/raetsch/lectures/amsa07

# Lecture Outline

- Basics
  - The Probabilistic Model
  - Some Probability and Statistics
  - Generative vs. Discriminative Approach
- Maximum Likelihood
  - Idea
  - Gaussian Distributions
  - ML for Gaussians

# Machine Learning Problem




. . .



natural +1

plastic -1

natural +1

plastic -1

?

## The 'Learning World':

**Data:** $\{(\mathbf{x}_{,}y_{)}\}_{=1}^{'}\mathbf{x}_{\in}^{d}, y_{\in}\{\pm 1\}$

**Objective:** Given new $\mathbf{x}$, predict $y$

**Problem:** The world is incomplete and uncertain!

# Machine Learning Problem



natural +1 | plastic -1 | natural +1 | ... | plastic -1 | ?

## The 'Learning World':

**Data:** $\{(\mathbf{x}_{,}y_{)}\}_{=1}^{,}\mathbf{x}_{\in}^{d}, y_{\in}\{\pm 1\}$

**Objective:** Given new $\mathbf{x}$, predict $y$

**Probabilistic Assumption:** All data is generated by the same hidden probabilistic source!

# Machine Learning Problem



natural +1 | plastic -1 | natural +1 | ... | plastic -1 | ?

## The 'Learning World':

**Data:** $\{(\mathbf{x}_{,}y_{)}\}_{=1}^{'}\mathbf{x}_{\in}^{d}, y_{\in}\{\pm 1\}$

**Objective:** Given new $\mathbf{x}$, predict $y$

**Probabilistic Assumption:** All data is generated by the same hidden probabilistic source!

# Probabilistic Model




. . .



natural +1

plastic -1

natural +1

plastic -1

?

## The 'Learning World':

**Data:** $\{(\mathbf{x},y)\}_{=1}^{'}\mathbf{x}_{\in}^{d}, y_{\in}\{\pm 1\}$

**Objective:** Find 'best' $y = h(\mathbf{x})$

**Assumption:** Sample $S_{=}(\mathbf{x}_1, y_1), \ldots, (\mathbf{x},y_{)} \sim p(\mathbf{x}, y)$

redProblem: $p(\mathbf{x}, y)$ is unknown!

# Learning Problem - Reformulated

MAX-PLANCK-GESELLSCHAFT

blue**Given:**

green**Hypothesis class:** $= \{h \mid h :^{d} \to \mathcal{Y}\}$ (e.g. $\mathcal{Y} = \{\pm 1\}$)

green**Loss:** $l(y, h(\mathbf{x}))$ (e.g. 0-1 loss $[y \neq h(\mathbf{x})]$)

green**Sample:** $S_{=}(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{,}y_{)} \sim p(\mathbf{x}, y)$

blue**Objective:** Minimize the true (expected) loss – "generalization error"

$$h^* = \min_{h \in red} L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\times}\, l(, h())}$$

blue**Problem:** Only have a greendata sample available, $p(\mathbf{x}, y)$ is unknown!

This is a very general setup! By varying  and $l$ one arrives at classification, regression, clustering.

One example is classification.

# Learning Problem - Reformulated

blue**Given:**

greenHypothesis class: $= \{h \mid h :^{d} \to \mathcal{Y}\}$ (e.g. $\mathcal{Y} = \{\pm 1\}$)

greenLoss: $l(y, h(\mathbf{x}))$ (e.g. 0-1 loss $[y \neq h(\mathbf{x})]$

greenSample: $S_{=}(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{,}y_{)} \sim p(\mathbf{x}, y)$

blue**Objective:** Minimize the true (expected) loss – "generalization error"

$$h^* = \min_{h \in red} L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\times}\, l(, h())}$$

blue**Problem:** Only have a greendata sample available, $p(\mathbf{x}, y)$ is unknown!

This is a very general setup! By varying  and $l$ one arrives at classification, regression, clustering.

One example is classification.

# Solution

blue**Given:**

green**Hypothesis class:** $= \{h \mid h :^{d} \rightarrow \{\pm 1\}\}$

green**Loss:** $l(y, h(\mathbf{x}))$ (e.g. 0-1 loss $[y \neq h(\mathbf{x})]$)

green**Sample:** $S_{=}(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{,}y_{)} \sim p(\mathbf{x}, y)$

blue**Objective:** Minimize the true (expected) loss – "generalization error"

$$h^* = \min_{h \in red} L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\times}\, l(, h())}$$

blue**Problem:** Only have a greendata sample available, $p(\mathbf{x}, y)$ is unknown!

red**Solution:** Estimate $p(\mathbf{x}, y)$ from $S$. Then minimize

$$\hat{h} = \hat{L}(h) := \hat{\mathbf{E}}_{\times}\, l(, h()).$$

# Questions

- Can we estimate any $p(\mathbf{x}, y)$ well algorithmically?
  - No free lunch!
- How does 'the approximation' of $p(\mathbf{x}, y)$ influence the generalization error of $\hat{h}$?
  - Occam's razor: good appproximation on sample + simple model $\Rightarrow$ good generalization.
  - Empirical distribution (ERM): statistical learning theory quantifies how generalization is influenced by choice of and $l$.
- Can we do estimation and optimization efficiently?
  - Sometimes, depending on our model.

# Least Squares

## Linear regression

$$y = \mathbf{w}^\top \mathbf{x} + \varepsilon$$

## Best Unbiased Linear Estimator

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$



## Gradient

$$\nabla_{\mathbf{w}} = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)\mathbf{x}_i^\top$$

## Solution

$$\mathbf{w}_{LS} = (X^\top X)^{-1} X^\top y$$

# Regularized Least Squares

## Linear regression

$$y = \mathbf{w}^\top \mathbf{x} + \varepsilon$$

## Bias the weights to be "small"

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \mathbf{w}^\top \mathbf{w}$$

## Gradient

$$\nabla_{\mathbf{w}} = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)\mathbf{x}_i^\top + \lambda \mathbf{w}$$

## Solution

$$\mathbf{w}_{RLS} = (X^\top X + \lambda \mathbf{I})^{-1} X^\top y$$

# Reading

- Probability books: ????.
- Machine Learning: ???.
- Theory of Classification: ??
- Online Resources
  - http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/pdf.html
  - http://www.inference.phy.cam.ac.uk/mackay/itila/book.html

# Probability

**Random Variable** is a function that maps outcomes of random experiments to numbers.

**Probability** is the fraction of times an event occurs.

**Lazy notation**

- $p(x)$ denotes the probability that random variable $X$ takes value $x$, i.e. $p(X = x)$.
- For continuous variables, $p(x)$ is technically called the probability density.

Note that these are not definitions!

# Joint, Marginal, Conditional

## Joint Probability

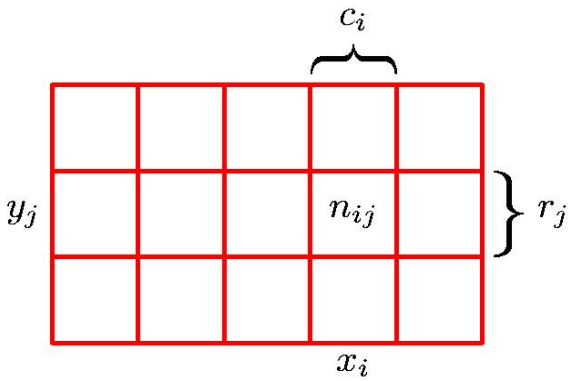
The entry of both values jointly.

$$p(X = x_i, Y = y_i) = \frac{n_{ij}}{N}$$

## Marginal Probability

The sum over a row or column.

$$p(X = x_i) = \frac{c_i}{N}$$




## Conditional Probability

The fraction of a row or column in a particular cell.

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

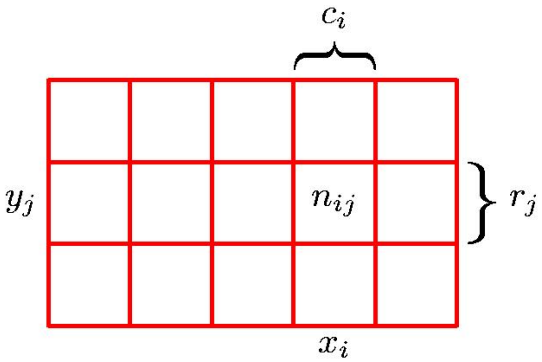
# Expectations

**Expectation** For discrete distributions, defined as the average value of $f(x)$ weighted by the probabilities

$$\mathbf{E}_X f(X) = \sum_i f(x_i) p(X = x_i).$$

**Conditional Expectation** Expectation w.r.t. the conditional distribution

$$\mathbf{E}_X[f(X)|Y] = \sum_i f(x_i) p(X = x_i|Y).$$



# Useful relations

**Factorization** If $X$ and $Y$ are independent events then

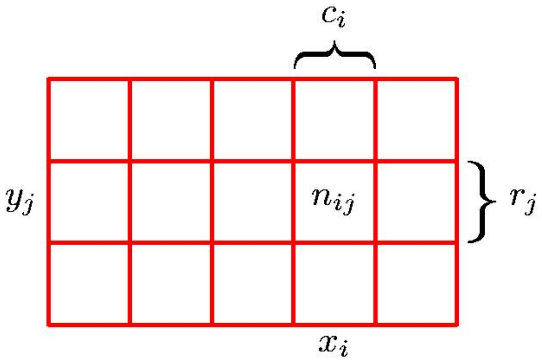
$$p(X = x_i, Y = y_j) = p(X = x_i)p(Y = y_j)$$

**Marginalization (Sum rule)**

$$p(X = x_i) = \sum_j p(X = x_i, Y = y_j).$$

**Product Rule**

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y).$$




**Bayes' Rule**

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

posterior $\propto$ class cond. $\times$ prior

# The Optimal Hypothesis

**Classification:** The function which minimizes the expected 0-1 loss is the one which assigns to each $\mathbf{x}$ the class with maximal posterior

$$h(\mathbf{x}) := \begin{cases} 1, & p(Y=1|\mathbf{x}) \geq p(Y=-1|\mathbf{x}) \\ -1, & \text{otherwise} \end{cases}$$

*(This easily extends to the multiclass case and to arbitrary loss-matrix.)*

**Regression:** The function which minimizes the mean square error is conditional expectation w.r.t the posterior

$$h(\mathbf{x}) := \mathbf{E}(y|\mathbf{x}).$$

What we want is thus $p(y|\mathbf{x})$.

# Generative vs Discriminative

## Generative approach

Models $p(\mathbf{x}, y) = p(\mathbf{x}|y)p(y)$. Uses Bayes' rule to infer

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}.$$

## Discriminative approach

Models $p(y|\mathbf{x})$ directly and takes max.

# Examples

## Generative

- Mixtures of Gaussians
- Hidden Markov Models
- Bayesian Networks
- Graphical Models
- . . .

## Discriminative

- SVM
- Least Squares Regression
- Regularized Least Squares Regression
- . . .

# Maximum Likelihood Approach

- Principled approach for density estimation when given data samples.
- Assume a parameterized class of densities:

$$p(x) \in \{p(x|\theta)|\theta \in \Theta\}.$$

- The likelihood function: models probability of observed data for each parameter:

$$l_x(\theta) = p(x|\theta).$$

- Idea: After seeing a data sample $D = x_1, x_2, \ldots x_n$ (usually iid) pick parameter which is most likely to have generated this sample (parameter estimation):

$$\theta^*_{ML} = \text{argmin}_\theta l_D(\theta).$$

# Maximum Likelihood Approach II

**IID Data:** $D = x_1, x_2, \ldots x_n$

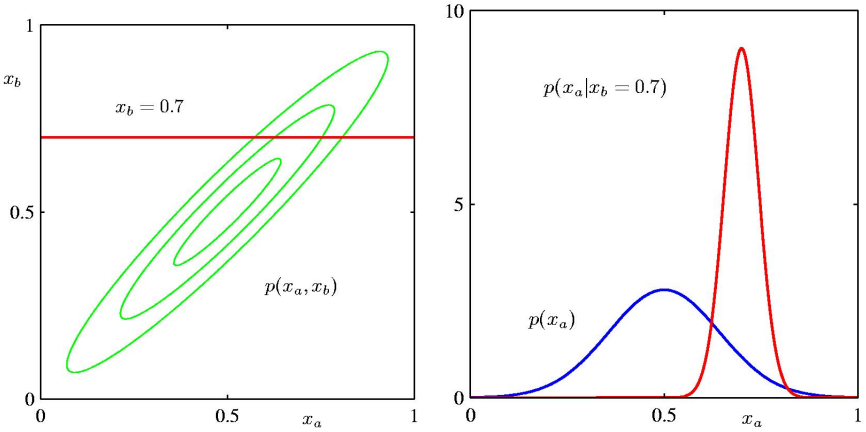
$$l_D(\theta) = \prod_i p(x_i|\theta).$$

## Negative Log Likelihood

Usually minimize $-\log p(D|\theta)$.

- Logarithm is monotonic
- Simplifies maths: IID assumption makes log a sum (derivatives termwise)
- Better numerical behaviour

# Gaussian Distribution (1)




## 1 dimensional

$$(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

## D dimensional

$$(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right)$$

# Gaussian Distribution (2)

## D dimensional

$$(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right)$$

## Negative Log Likelihood

$$-\log p(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{D}{2}\log 2\pi$$

## Important Properties

- All marginals and conditionals are again Gaussian.
- Sufficient statistics are mean and variance.
- Part of the exponential family distributions.

# ML for Gaussian Posteriors

## Likelihood Function

$$p(y|\mathbf{x}) = p(y|\mathbf{x}, \theta); \quad l_{\mathbf{x}}(\theta) = p(y|\mathbf{x}, \theta) \sim (\mu(\theta), \sigma(\theta)^2)$$

## Linear Regression with Gaussian Noise

$$y = \mathbf{w}^\top \mathbf{x} + \varepsilon,$$

where $\varepsilon \sim (0, \sigma^2)$.

## Gaussian noise

Using the fact that $\varepsilon$ is Gaussian, then $y_i \sim (\mathbf{w}^\top \mathbf{x}_i, \sigma^2)$, and hence

$$p(y|\mathbf{x}, \mathbf{w}) = \prod_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i, \sigma^2)$$

# Optimization

## Maximum Likelihood with Gaussian Noise

$$p(y|\mathbf{x}, \mathbf{w}) = \prod_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i, \sigma^2)$$

## Negative Log Likelihood

$$\min_{\mathbf{w}} -\log p(y|\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

## Least Squares

Observe that the negative log likelihood is the least squares problem.

## Solution

$$\mathbf{w}_{ML} = (X^\top X)^{-1} X^\top y$$

# Probabilistic Model

## Maximum Likelihood

We have minimized the negative log likelihood, $-\log p(y|\mathbf{x}, \mathbf{w})$ to obtain the weight vector $\mathbf{w}_{ML}$.

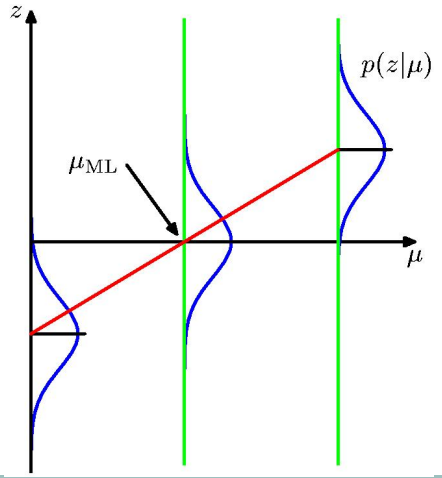
## Recall

We have a probabilistic model, and hence we can make a predictive distribution for a new value $\mathbf{x}_t$.

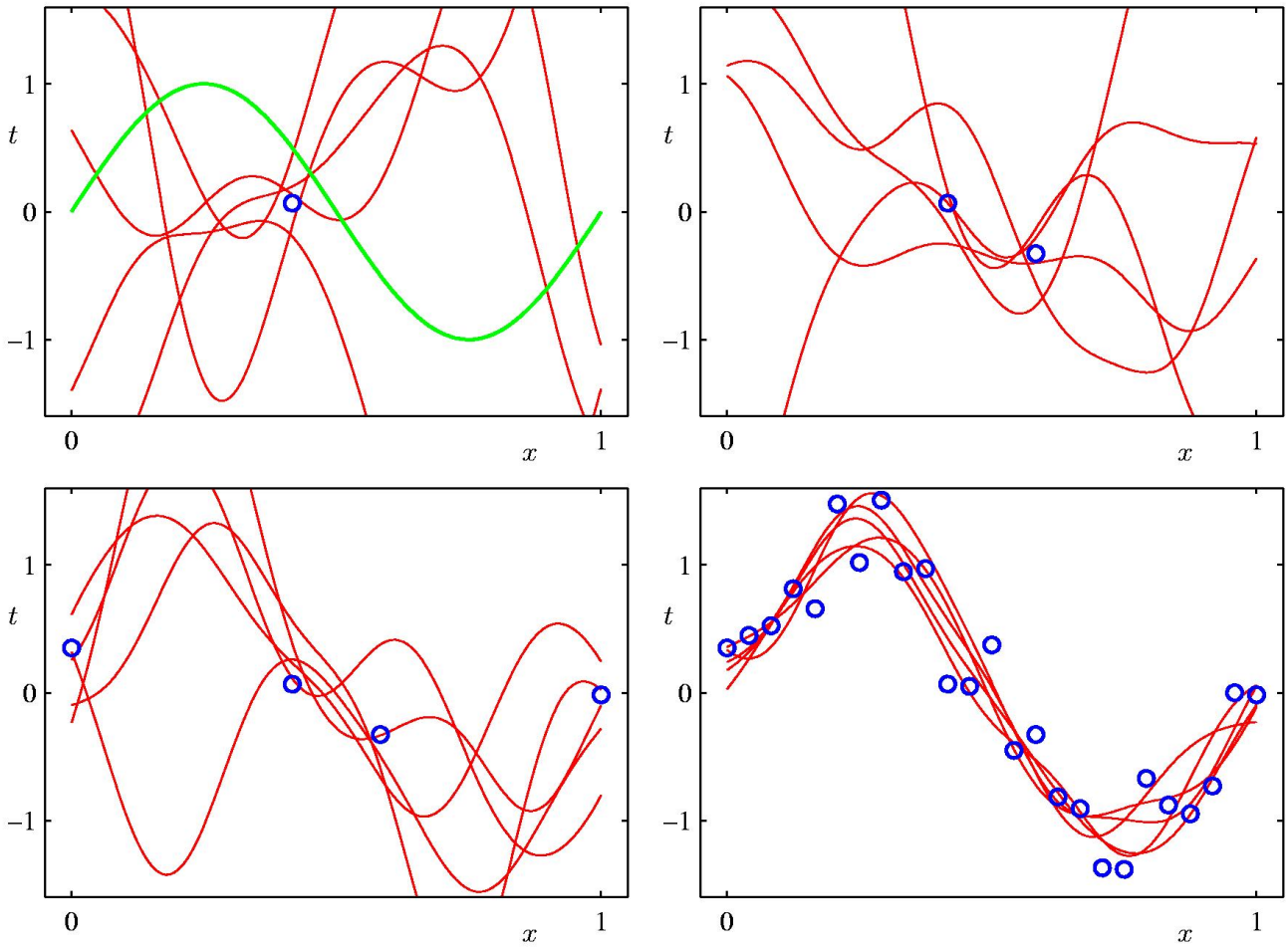
## In other words

For each test example $\mathbf{x}_t$, we get a distribution

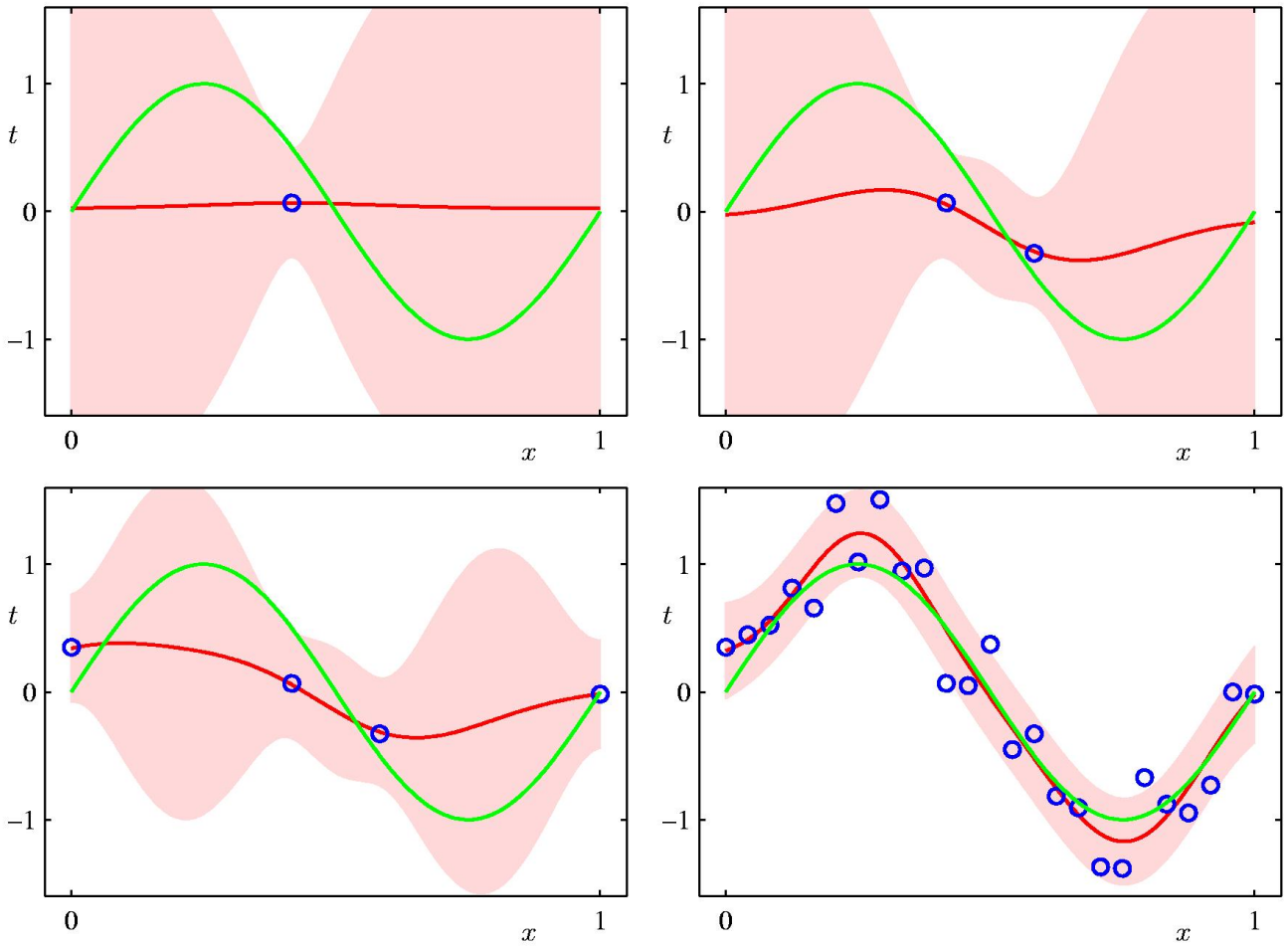
$$y_t \sim (\mathbf{w}_{ML}^\top \mathbf{x}_t, \sigma^2).$$



# Regression: Samples



# Regression: Variance



# Prior

## Bayes rule

$$p(\mathbf{w}|\mathbf{x}, y) = \frac{p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w})}{p(y|\mathbf{x})}$$

posterior $\propto$ likelihood $\times$ parameter prior

## Using background information

We can include prior knowledge about how we would like the weight vector, for example that it is multivariate Gaussian:

$$\mathbf{w} \sim (0, \Sigma^{-1}).$$

i.e.

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{w}^\top \mathbf{\Sigma}^{-1}\mathbf{w}\right).$$

# Maximum a Posteriori

## Idea

Instead of finding $\mathbf{w}$ that maximizes likelihood $p(y|\mathbf{x}, \mathbf{w})$, we find the $\mathbf{w}$ that maximizes the posterior $p(\mathbf{w}|\mathbf{x}, y)$.

## Posterior

$$p(\mathbf{w}|\mathbf{x}, y) \propto \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top X)^\top (y - \mathbf{w}^\top X)\right) \exp\left(-\frac{1}{2}\mathbf{w}^\top \mathbf{\Sigma}^{-1}\mathbf{w}\right)$$

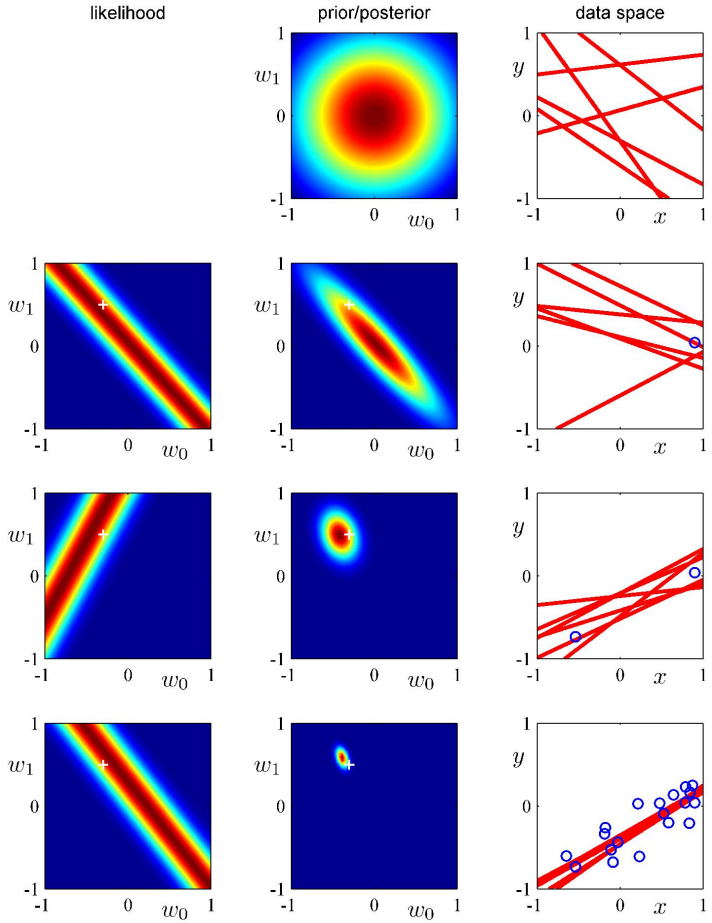
## Negative Log Posterior

$$-\log p(\mathbf{w}|\mathbf{x}, y) = -\frac{1}{2\sigma^2}(y - \mathbf{w}^\top X)^\top (y - \mathbf{w}^\top X) - -\frac{1}{2}\mathbf{w}^\top \mathbf{\Sigma}^{-1}\mathbf{w}$$

## Regularized Least Squares

The negative log posterior is the regularized least squares problem.

# How MAP Works



# Have Seen

- Maximum Likelihood with Gaussian posteriors for linear regression with Gaussian noise is the least squares solution.
- Maximum a posteriori with a Gaussian prior for the same problem is the regularized least squares solution.

# Binary Classification

The posterior probability for the positive class is given by

$$p(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-a)} = \text{sig}(a)$$

where

$$a = \ln \frac{p(\mathbf{x}|Y = 1)p(Y = 1)}{p(\mathbf{x}|Y = -1)p(Y = -1)}.$$

## Sigmoid Function

$$\text{sig}(a) = \frac{1}{1 + \exp(-a)}$$

("S" shaped).

# ML for Class Conditionals

**Maximum Likelihood:** Assume Gaussian class conditional densities with equal covariances and solve maximum likelihood with constraint

$$p(\mathbf{x}|Y=1) \sim (\mu_+, \Sigma^{-1}); \quad p(\mathbf{x}|Y=-1) \sim (\mu_-, \Sigma^{-1}).$$

**Linear Term in $\mathbf{x}$:** From the results in the previous slide, the posterior is $p(Y=1|\mathbf{x}) = \mathsf{sig}(\mathbf{w}^\top x + w_0)$, where

$$\mathbf{w} = \Sigma^{-1}(\mu_+ - \mu_-)$$

$$w_0 = -\frac{1}{2}\mu_+^\top \Sigma^{-1}\mu_+ + \frac{1}{2}\mu_-^\top \Sigma^{-1}\mu_- + \ln \frac{p(Y=1)}{p(Y=-1)}$$

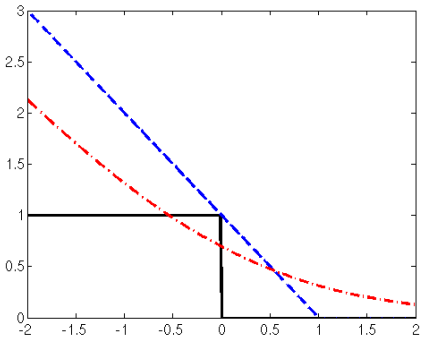
**Linear Classifier:**

$$\begin{aligned} w^* &= \mathsf{argmax}_w \, \mathsf{sig}(y_i(\mathbf{w}^\top x + w_0)) \\ &= \mathsf{argmin}_w (1 + \exp(-y_i(\mathbf{w}^\top x + w_0))). \end{aligned}$$

# Logistic Regression

- Logistic regression solves a classification task.
- It assumes a sigmoid function acting on a linear model, i.e. it assumes a logistic loss

$$\ell(f(\mathbf{x}_i), y_i) := \log(1 + \exp(-y_i f(\mathbf{x}_i)))$$




- Can solve for the maximum likelihood (unregularized) or maximum a posteriori (regularized) weights.

# Summary

- Probabilistic view provides different tools for analysis
- Bayes rule gives the relationship between the class conditionals and the posterior.
- From the probabilistic model, one can do both generative and discriminative methods via maximum likelihood.
- Maximum likelihood approach for posteriors or class conditionals recovers least squares regression and logistic regression.
- MAP can be seen as regularized ML estimation.