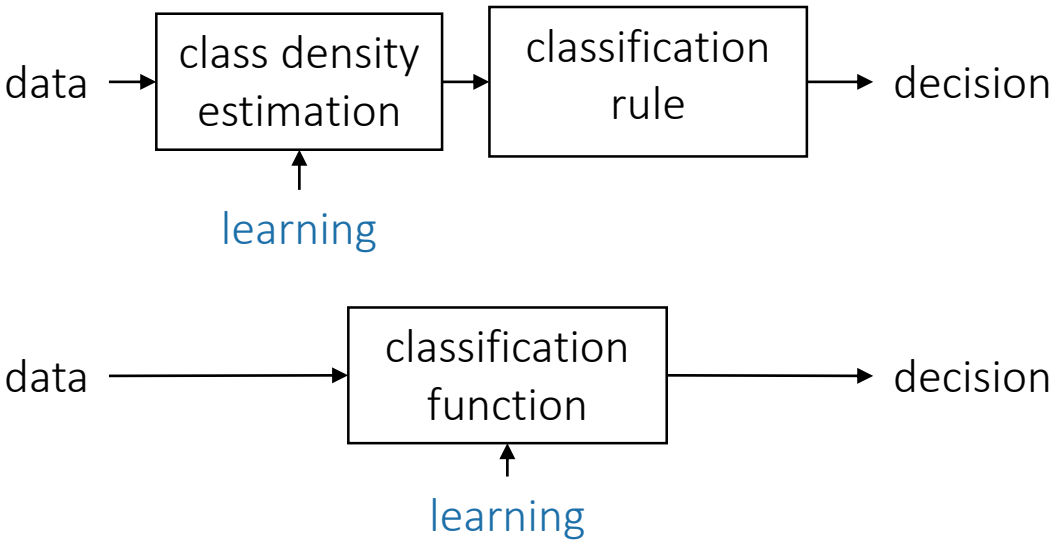
Logistic Regression

Two Worlds: Probabilistic & Algorithmic

We know two conceptual approaches to classification:



Bayes Classifier
Probabilistic classifier with a generative setup based on class density models
Bayes (Gauss), Naïve Bayes

"Direct" Classifiers
Find best parameter (e.g. $\vec{w}$) with respect to a specific loss function measuring misclassification
Perceptron, SVM, Tree, ANN

Can we have a probabilistic classifier with a modelling focus on *classification*?

Advantages of Both Worlds

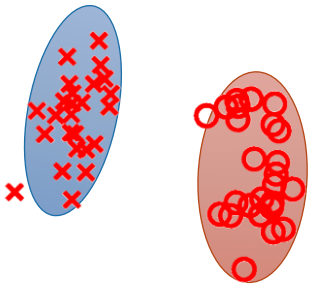
- Posterior distribution has advantages over classification label:
  - Asymmetric risks: need classification probability
  - Classification certainty: Indicator if decision in unsure

- Algorithmic approach with direct learning has advantages:
  - Focus of modelling power on correct classification where it counts
  - Easier decision line interpretation

- Combination?

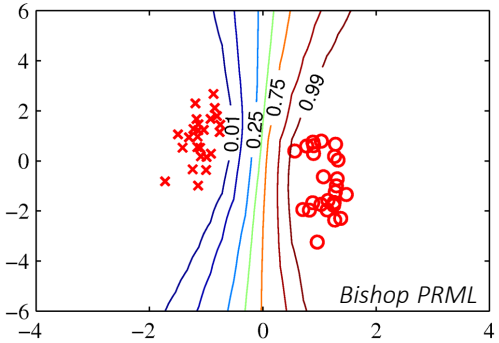
Discriminative Probabilistic Classifier

Bayes Classifier

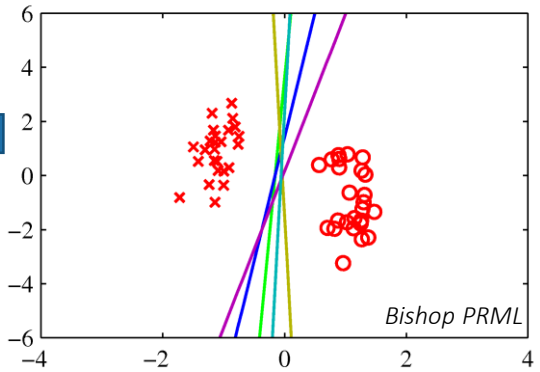


$P(x|C_1)$ $P(x|C_2)$

Discriminative Probabilistic Classifier



Linear Classifier



$$g(\vec{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

$$P(C_2|x) \propto P(x|C_2)P(C_2)$$

Towards a “Direct” Probabilistic Classifier

- Idea 1: Directly learn a posterior distribution

  For classification with the Bayes classifier, the posterior distribution is relevant. We can directly estimate a model of this distribution. We know from Naïve Bayes that we can probably expect a good performance from the posterior model.

- Idea 2: Extend linear classification with probabilistic interpretation

  The linear classifier outputs a distance to the decision plane. We can use this value and interpret it probabilistically: “The further away, the more certain”

Logistic Regression

The *Logistic Regression* will implement both ideas: It is a model of a posterior class distribution for classification and can be interpreted as a probabilistic linear classifier. But it is a fully probabilistic model, not only a “post-processing” of a linear classifier.

It extends the hyperplane decision idea to Bayes world

- Direct model of the posterior for classification
  - Probabilistic model (classification according to a probability distribution)
  - Discriminative model (models posterior rather than likelihood and prior)
- Linear model for classification
  - Simple and accessible (we can understand that)
  - We can study the relation to other linear classifiers, i.e. SVM

History of Logistic Regression

- Logistic Regression is a very “old” method of statistical analysis and in widespread use, especially in the traditional statistical community (not machine learning).

  1957/58, Walker, Duncan, Cox

- A method more often used to study and identify explaining factors rather than to do individual prediction.

  Statistical analysis vs. prediction focus of modern machine learning

  Many medical studies of risk factors etc. are based on logistic regression

Statistical Data Models

We do not know $P(x,y)$ but we can assume a certain form.

---> This is called a data model.

Simplest form besides constant (one prototype) is a linear model.

$$Lin_w(x) = \sum_{i=1}^{d} w_i x_i + w_0 = \langle w, x \rangle + w_0 = \vec{w}^T \vec{x} + w_0$$

$$\tilde{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}, \; \tilde{w} = \begin{bmatrix} w_0 \\ w \end{bmatrix}$$

$$\Rightarrow Lin_w(x) = \langle w, x \rangle + w_0 = \langle \tilde{w}, \tilde{x} \rangle$$

- Linear Methods:

  Classification: Logistic Regression (no typo!)

  Regression: Linear Regression

Repetition: Linear Classifier

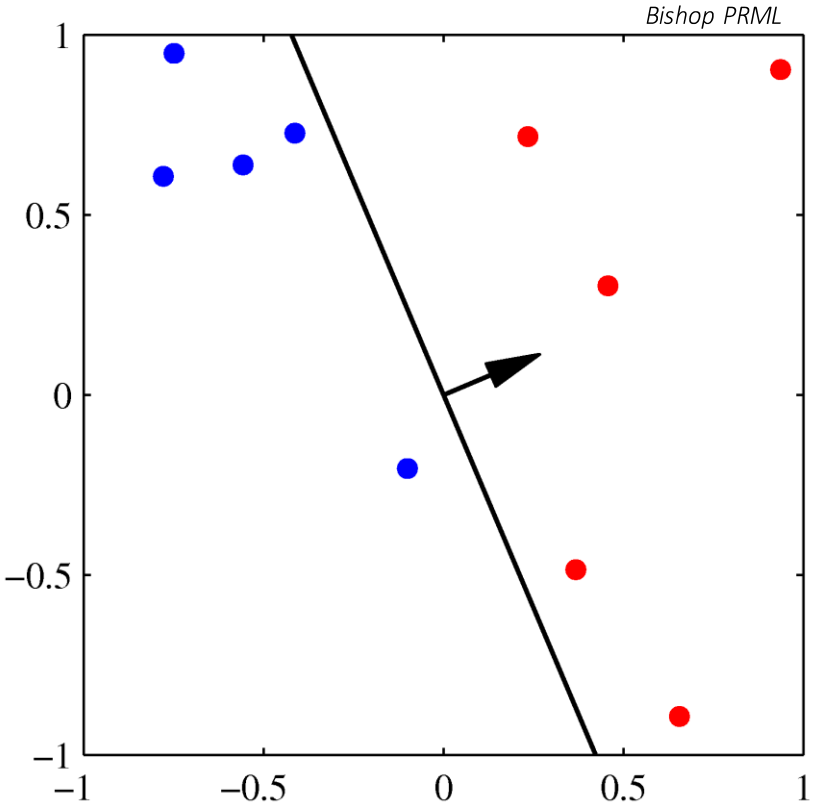
Linear classification rule:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

$g(\boldsymbol{x}) \geq 0 \Rightarrow$ ●

$g(\boldsymbol{x}) < 0 \Rightarrow$ ●

Decision boundary is a a *hyperplane*

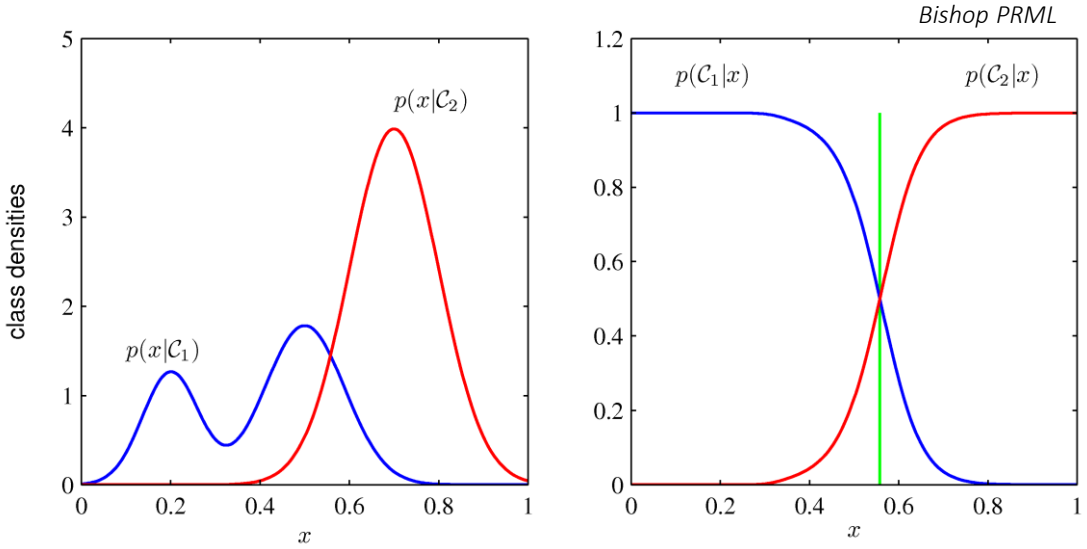


Repetition: Posterior Distribution

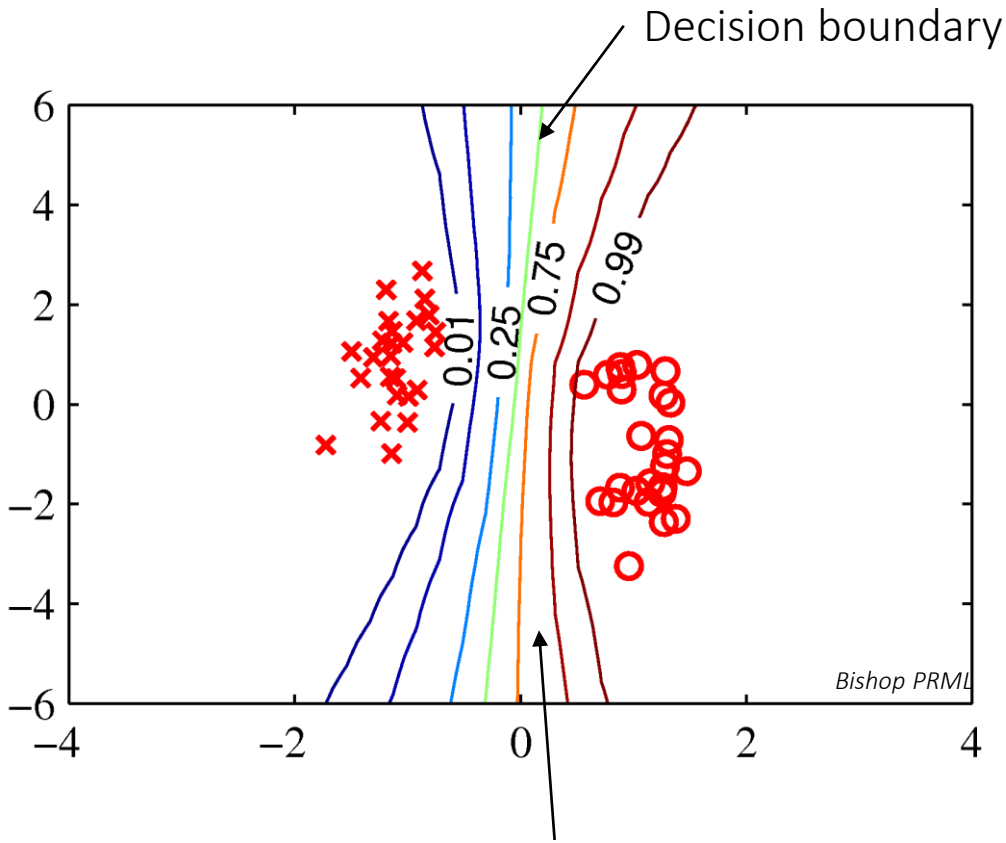
- Classification with Posterior distribution: Bayes

  Based on class densities and a prior

$$P(C_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_1)P(C_1)}{p(\boldsymbol{x}|C_1)P(C_1) + p(\boldsymbol{x}|C_2)P(C_2)} \qquad P(C_2|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_2)P(C_2)}{p(\boldsymbol{x}|C_1)P(C_1) + p(\boldsymbol{x}|C_2)P(C_2)}$$



Combination: Discriminative Classifier



Probabilistic interpretation of classification output: ~distance to separation plane

Notation Changes

- We work with two classes

  Data with (numerical) feature vectors $\vec{x}$ and labels $\boldsymbol{y} \in \{\mathbf{0}, \mathbf{1}\}$

  We do not use the notation of Bayes with ω anymore. We will need the explicit label value of $\boldsymbol{y}$ in our models later.

- Classification goal: infer the best class label $\{\mathbf{0}\ \boldsymbol{or}\ \mathbf{1}\}$ for a given feature point

$$y^* = \arg\max_{y \in \{0,1\}} P(y|\boldsymbol{x})$$

- All our modeling focuses only on the posterior of having class 1:

$$P(y = 1\,|\boldsymbol{x}\,)$$

  - Obtaining the other is trivial: $P(y = 0\,|\boldsymbol{x}\,) = 1 - P(y = 1\,|\boldsymbol{x}\,)$

Parametric Posterior Model

We need a model for the posterior distribution, depending on the feature vector (of course) and neatly parameterized.

$$P(y = 1 \mid \boldsymbol{x}, \boldsymbol{\theta}) = f(\boldsymbol{x};\ \boldsymbol{\theta})$$

The linear classifier is a good starting point. We know its parametrization very well:

$$g(\boldsymbol{x};\ \boldsymbol{w}, w_0) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

We thus model the posterior as a function of the linear classifier:

$$P(y = 1 |\ \boldsymbol{x}, \boldsymbol{w}, w_0) = f(\boldsymbol{w}^T \boldsymbol{x} + w_0)$$

Posterior from classification result: “scaled distance“ to decision plane

Logistic Function

To use the *unbounded* distance to the decision plane in a probabilistic setup, we need to map it into the interval $[0, 1]$

This is very similar as we did in neural nets: *activation function*

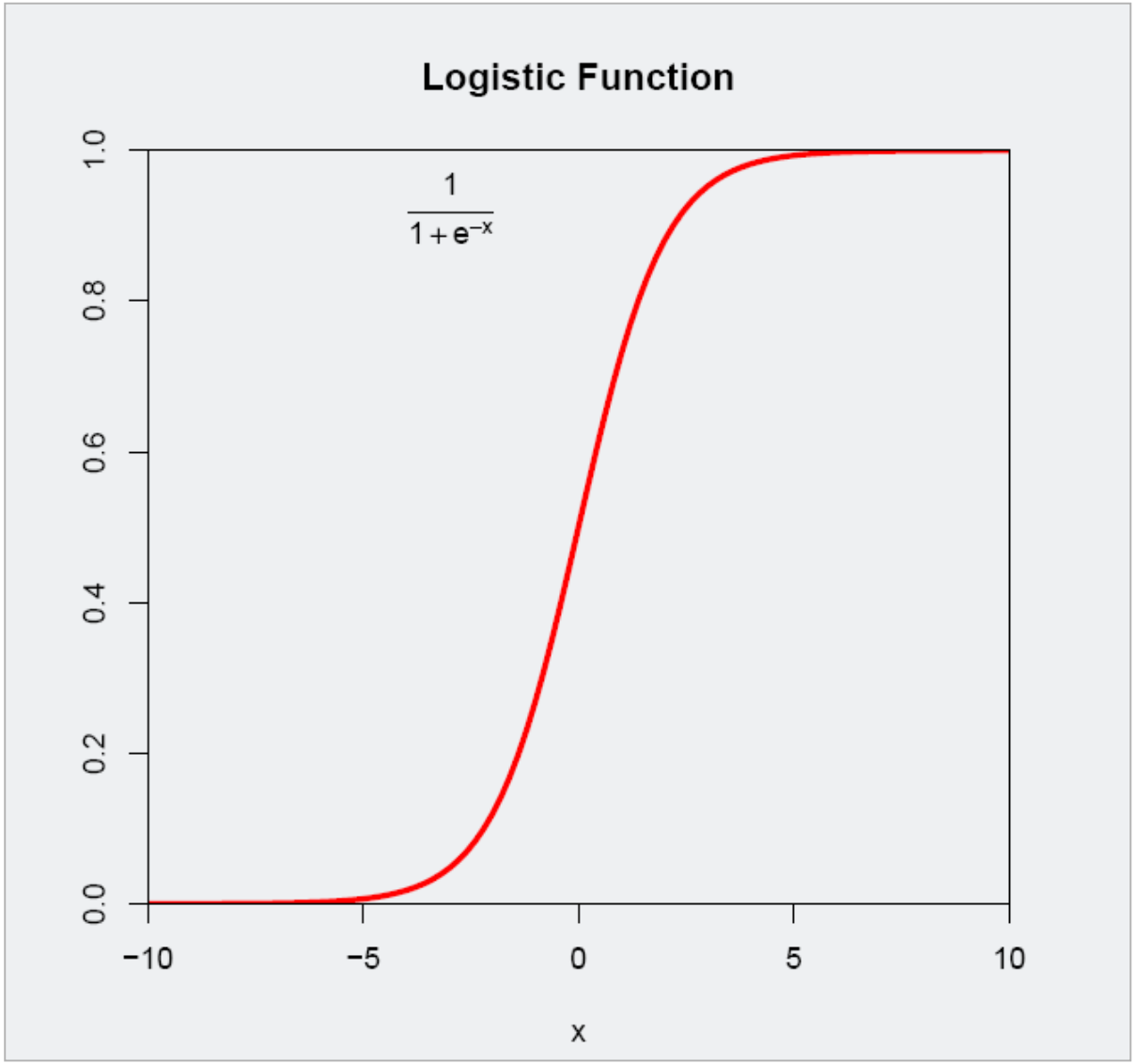
The *logistic function* $\sigma(x)$ squashes a value $x \in \mathbb{R}$ to $[0, 1]$

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$

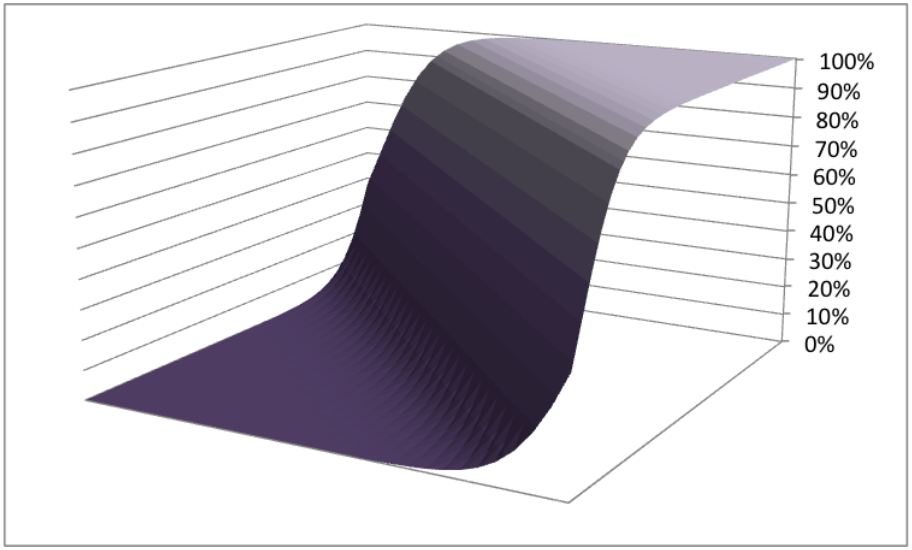
The logistic function is a smooth, soft threshold

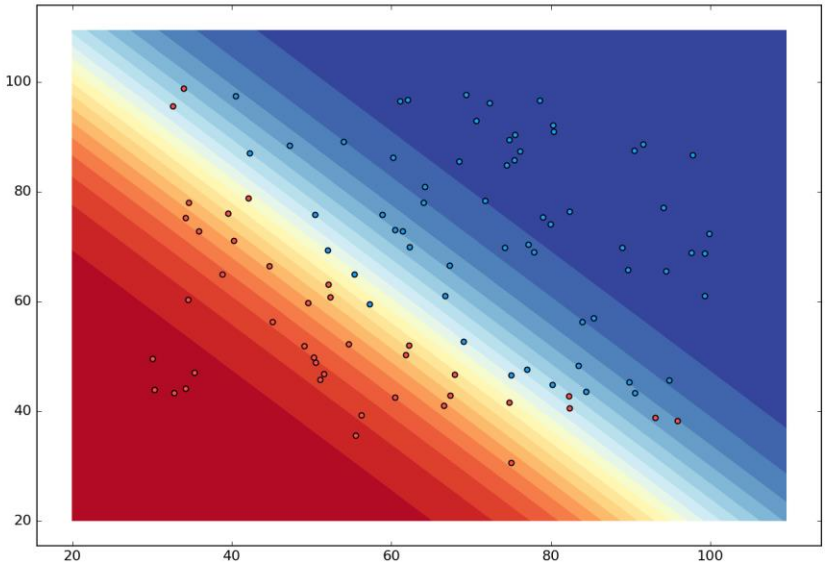
$$\sigma(x) \to 1 \quad x \to \infty$$
$$\sigma(x) \to 0 \quad x \to -\infty$$
$$\sigma(0) = \frac{1}{2}$$

The Logistic Function



The Logistic “Regression”





The Logistic Regression Posterior

We model the posterior distribution for classification in a two-classes-setting by applying the logistic function to the linear classifier:

$$P(y = 1|\ x) = \sigma\big(g(x)\big)$$

$$P(y = 1|\ \boldsymbol{x}, \boldsymbol{w}, w_0) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0) = \frac{1}{1 + e^{-(\boldsymbol{w}^T\boldsymbol{x} + w_0)}}$$

This a location-dependent model of the posterior distribution, parametrized by a linear hyperplane classifier.

Logistic Regression is a Linear Classifier

The logistic regression posterior leads to a linear classifier:

$$P(y = 1|\ \boldsymbol{x}, \boldsymbol{w}, w_0) = \frac{1}{1 + \exp(-(\boldsymbol{w}^T \boldsymbol{x} + w_0))}$$

$$P(y = 0|\ \boldsymbol{x}, \boldsymbol{w}, w_0) = 1 - P(y = 1|\ \boldsymbol{x}, \boldsymbol{w}, w_0)$$

$P(y = 1|\ \boldsymbol{x}, \boldsymbol{w}, w_0) > \frac{1}{2}$ $\Rightarrow$ $y = 1$ classification; $y = 0$ otherwise

Classification boundary is at: $P(y = 1|\ \boldsymbol{x}, \boldsymbol{w}, w_0) = \frac{1}{2}$

$$\Rightarrow \quad \frac{1}{1 + \exp(-(\boldsymbol{w}^T \boldsymbol{x} + w_0))} = \frac{1}{2} \qquad \Rightarrow \quad \boxed{\boldsymbol{w}^T \boldsymbol{x} + w_0 = 0}$$

Classification boundary is a hyperplane

Interpretation: Logit

Is the choice of the logistic function justified?

- Yes, the *logit* is a linear function of our data:

  Logit: log of the *odds ratio:* $\ln \frac{p}{1-p}$

$$\ln \frac{P(y=1|\boldsymbol{x})}{P(y=0|\boldsymbol{x})} = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

The linear function (~distance from decision plane) directly expresses our classification certainty, measured by the "odds ratio":
double distance ↔ squared odds
e.g. $3:2 \rightarrow 9:4$

- But other choices are valid, too

  They lead to other models than logistic regression, e.g. probit regression

  → *Generalized Linear Models (GLM)*

$$E[y] = f^{-1}(\boldsymbol{w}^T \boldsymbol{x} + w_0)$$

The Logistic Regression

- So far we have made no assumption on the data!
- We can get $r(x)$ from a generative model or model it directly as function of the data (discriminative)

Logistic Regression:

Model: The logit $r(x) = \log \frac{P(y=1|x)}{P(y=0|x)} = \log \frac{p}{1-p}$

is a linear function of the data

$$r(x) = \log \frac{p}{1-p} = \sum_{i=1}^{d} w_i x_i + w_0 = \langle \tilde{w}, \tilde{x} \rangle$$

<= > $$P(y=1 \mid x) = \sigma(\langle \tilde{w}, \tilde{x} \rangle) = \frac{1}{1+\exp(-\langle \tilde{w}, \tilde{x} \rangle)}$$

Training a Posterior Distribution Model

The posterior model for classification requires training. Logistic regression is not just a post-processing of a linear classifier. Learning of good parameter values needs be done with respect to the probabilistic meaning of the posterior distribution.

- In the probabilistic setting, learning is usually estimation

  We now have a slightly different situation than with Bayes: We do not need class densities but a good *posterior distribution*.

- We will use Maximum Likelihood and Maximum-A-Posteriori estimates of our parameters $\boldsymbol{w}, w_0$

Later: This also corresponds to a cost function of obtaining $\boldsymbol{w}, w_0$

Maximum Likelihood Learning

The Maximum Likelihood principle can be adapted to fit the posterior distribution (discriminative case):

- We choose the parameters $\boldsymbol{w}, w_0$ which maximize the *posterior distribution* of the training set $\boldsymbol{X}$ with labels $\boldsymbol{Y}$:

$$\boldsymbol{w}, w_0 = \arg\max_{\boldsymbol{w}, w_0} P(\ \mathrm{Y}\ \mid\ X; \boldsymbol{w}, w_0)$$

$$= \arg\max_{\boldsymbol{w}, w_0} \prod_{\boldsymbol{x} \in X} P(y \mid\ \boldsymbol{x}; \boldsymbol{w}, w_0) \quad \text{(iid)}$$

$$P(y \mid \boldsymbol{x}; \boldsymbol{w}, w_0) = \quad P(\ y = 1 \mid\ \boldsymbol{x}; \boldsymbol{w}, w_0)^y \ \ P(\ y = 0 \mid\ \boldsymbol{x}; \boldsymbol{w}, w_0)^{1-y}$$

Logistic Regression: Maximum Likelihood Estimate of w (1)

To simplify the notation we use $\boldsymbol{w}, \boldsymbol{x}$ instead of $\boldsymbol{w}, w_0$

With $P\left(y=1\middle|x\right)=\sigma\left(w^T x\right)$ and $P\left(y=0\middle|x\right)=1-\sigma\left(w^T x\right)$

$$\Rightarrow P\left(y\middle|x\right)=P\left(y=1\middle|x\right)^y P\left(y=0\middle|x\right)^{1-y}=p^y(1-p)^{1-y}$$

The discriminative (log) likelihood function for our data

$$P\left(Y\middle|X\right) = \prod_{i=1}^{N} P\left(y_i\middle|x_i\right) = \prod_{i=1}^{N} p_i^{y_i}(1-p_i)^{1-y_i}$$

$$\log P\left(Y\middle|X\right) = \sum_{i=1}^{N} y_i \log\left(p_i\right)+\left(1-y_i\right)\log\left(1-p_i\right)$$

"cross-entropy" cost function

$$= \sum_{i=1}^{N} y_i \log\left(\frac{p_i}{1-p_i}\right)+\log\left(1-p_i\right)$$

Maximum Likelihood Estimate of w (2)

log-likelihood function continued

$$\log L(Y,X) \equiv \log P(Y|X) = \sum_{i=1}^{N} y_i \log\left(\frac{p_i}{1-p_i}\right) + \log(1-p_i)$$

Remember $p_i = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$ and linear Logit $\log \frac{p_i}{1-p_i} = w^T x$

$$\log L(Y,X) = \sum_{i=1}^{N} y_i w^T x_i - \log\left(1+e^{w^T x_i}\right)$$

Maximize the log-likelihood function with respect to W

$$\frac{\partial}{\partial w} \log L(Y,X) \stackrel{!}{=} 0$$

Maximum Likelihood Estimate of $\boldsymbol{w}$ (3)

$$\frac{\partial}{\partial w}\log L(Y,X) = \frac{\partial}{\partial w}\sum_{i=1}^{N} y_i w^T x_i - \log\left(1+e^{w^T x_i}\right)$$

$$= \sum_{i=1}^{N} y_i x_i^T - \frac{e^{w^T x_i}}{1+e^{w^T x_i}} x_i^T$$

Derivative of a Dot Product

Gradient operator

$$\frac{\partial}{\partial \boldsymbol{w}} = \nabla_{\mathbf{w}} = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_d}\right]$$

$$\frac{\partial}{\partial \boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{x} = \left[\frac{\partial}{\partial w_1} \boldsymbol{w}^T \boldsymbol{x}, \frac{\partial}{\partial w_2} \boldsymbol{w}^T \boldsymbol{x}, \dots, \frac{\partial}{\partial w_d} \boldsymbol{w}^T \boldsymbol{x}\right]$$

Per component

$$\frac{\partial}{\partial w_i} \boldsymbol{w}^T \boldsymbol{x} = \frac{\partial}{\partial w_i} \sum_{k=0}^{d} w_k x_k = x_i$$

Final derivative

$$\frac{\partial}{\partial \boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{x} = [x_1, x_2, \dots, x_d] = \boldsymbol{x}^T$$

Maximum Likelihood Estimate of $\boldsymbol{w}$ (3)

$$\frac{\partial}{\partial w}\log L(Y,X) = \frac{\partial}{\partial w}\sum_{i=1}^{N} y_i w^T x_i - \log\left(1+e^{w^T x_i}\right)$$

$$= \sum_{i=1}^{N} y_i x_i^T - \frac{e^{w^T x_i}}{1+e^{w^T x_i}} x_i^T$$

$$\frac{e^{w^T x_i}}{1+e^{w^T x_i}} = \frac{1}{1+e^{-w^T x_i}}$$

$$= \sum_{i=1}^{N} \left(y_i - \sigma\left(w^T x_i\right)\right) x_i^T \overset{!}{=} 0$$

- Non-linear equation in w : no closed form solution.
- The function $\mathrm{Log}\, L$ is concave therefore a unique maximum exists.

Iterative Reweighted Least Squares

The concave $\log P(\boldsymbol{Y}|\boldsymbol{X})$ can be maximized iteratively with the Newton-Raphson algorithm: *Iterative Reweighted Least Squares*

$$\boldsymbol{w}^{n+1} \leftarrow \boldsymbol{w}^{n} - \boldsymbol{H}^{-1} \frac{\partial}{\partial \boldsymbol{w}} (\ln P(\boldsymbol{Y}|\boldsymbol{X}; \boldsymbol{w}^{n}))$$

Derivatives and evaluation always with respect to $\boldsymbol{w}^{n}$

Hessian: Concave Likelihood

$$\boldsymbol{H} = \frac{\partial^2}{\partial \boldsymbol{w} \partial \boldsymbol{w}^T} \ln P(Y|X)$$

We use an old trick to keep it simple:

$$\boldsymbol{w} \coloneqq \begin{bmatrix} w_0 \\ \boldsymbol{w} \end{bmatrix}, \qquad \boldsymbol{x} \coloneqq \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}$$

$$\frac{\partial}{\partial \boldsymbol{w}} \left(\frac{\partial}{\partial \boldsymbol{w}^T} \ln P(Y|X) \right) = -\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T \sigma(\boldsymbol{w}^T \boldsymbol{x}_i) \big(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)\big) = -\boldsymbol{X} \boldsymbol{S} \boldsymbol{X}^T$$

The Hessian is negative definite:

- The sample covariance matrix $\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T$ is positive definite
- $\sigma(\boldsymbol{w}^T \boldsymbol{x}_i)\big(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)\big)$ is always positive

The optimization problem is said to be *convex* and has thus a optimal solution which can be iteratively calculated.

Iterative Reweighted Least Squares

The concave $\log P(\boldsymbol{Y}|\boldsymbol{X})$ can be maximized iteratively with the Newton-Raphson algorithm: *Iterative Reweighted Least Squares*

$$\boldsymbol{w}^{n+1} \leftarrow \boldsymbol{w}^{n} - \boldsymbol{H}^{-1} \frac{\partial}{\partial \boldsymbol{w}} (\ln P(\boldsymbol{Y}|\boldsymbol{X}; \boldsymbol{w}^{n}))$$

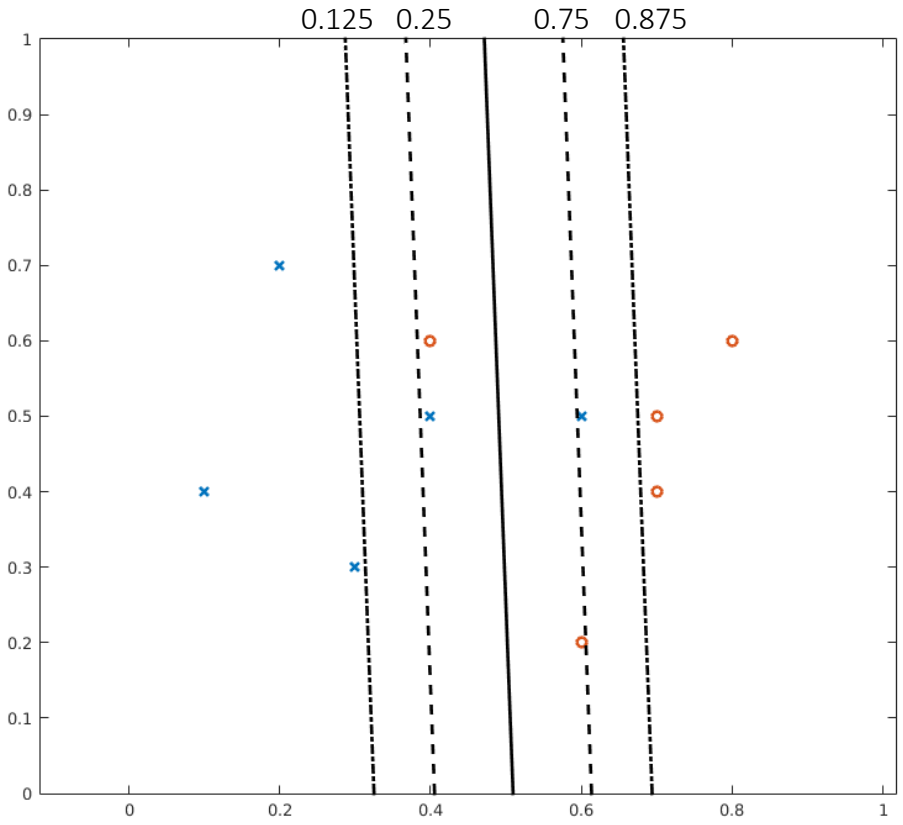
Derivatives and evaluation always with respect to $\boldsymbol{w}^{n}$

Method results in an iteration of *reweighted* least-squares steps

$$w^{n+1} = (XSX^{T})^{-1} X\ S\ \ z$$
$$z = X^{T} w^{n} + S^{-1}\left(Y - P(w^{n})\right)$$

- Weighted least-squares with $\boldsymbol{z}$ as target: $(XSX^{T})^{-1} X\ S\ \ z$
- $\boldsymbol{z}$: adjusted responses (updated every iteration)
- Matrix : $w^{n+1} = w^{n} - (XSX^{T})^{-1}(Y - \sigma\left(w^{n^{T}} X\right))X^{T}$

Example: Logistic Regression



Solid line: classification ($p = 0.5$)

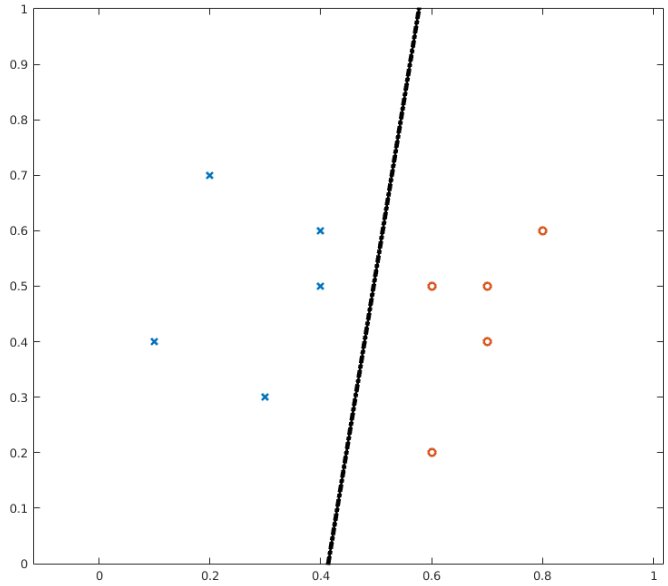
Dashed lines: $p = 0.25$, $p = 0.75$ lines

Probabilistic result: posterior of classification *everywhere*

The posterior probability decays/increases with distance to the decision boundary

Linearly Separable

- Maximum Likelihood learning is problematic in the linearly separable case: w diverges in length
  → leads to classification with *infinite* certainty
- Classification is still right but posterior estimate is not



Prior Assumptions

- Infinitely certain classification is likely an estimation artefact:
  We do not have enough training samples
  → maximum likelihood estimation leads to problematic results
- Solution: MAP estimate with prior assumptions on $\boldsymbol{w}$

$$P(w) = N(w|0, \sigma^2 I)$$

Smaller $\boldsymbol{w}$ are preferred (*shrinkage*)

$$P(y|\boldsymbol{x}, \boldsymbol{w}, w_0) = p^y(1-p)^{1-y}$$

Likelihood model is unchanged

$$\boldsymbol{w}, w_0 = \arg\max_{\boldsymbol{w}, w_0} P(\mathrm{Y}|X; \boldsymbol{w}, w_0\ )P(\boldsymbol{w})$$

$$= \arg\max_{\boldsymbol{w}, w_0} P(\boldsymbol{w}) \prod_{\boldsymbol{x} \in X} P(y|\boldsymbol{x}, \boldsymbol{w}, w_0\ )$$

MAP Learning

$$\ln P(\boldsymbol{w}) \prod_{\boldsymbol{x} \in X} P(y|\boldsymbol{x}, \boldsymbol{w}, w_0) =$$

$$\sum (\, y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - \ln(1 + \exp(\boldsymbol{w}^T \boldsymbol{x}_i + w_0)) \,) - \frac{1}{2\sigma^2} \|\boldsymbol{w}\|^2$$

We need: $\frac{\partial}{\partial \boldsymbol{w}} \|\boldsymbol{w}\|^2 = 2\boldsymbol{w}^T$

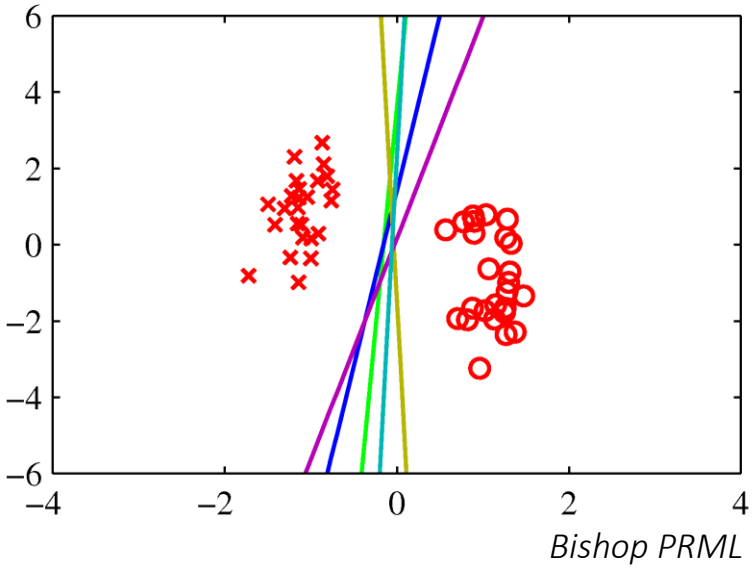
$$\frac{\partial}{\partial \boldsymbol{w}} \ln P(Y|X) = \sum_i (\, \big(y_i - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i + w_0)\big) \boldsymbol{x}_i^T \,) - \frac{1}{\sigma^2} \boldsymbol{w}^T \stackrel{!}{=} 0$$

- Iterative solution: Newton-Raphson
- Prior enforces a *regularization*

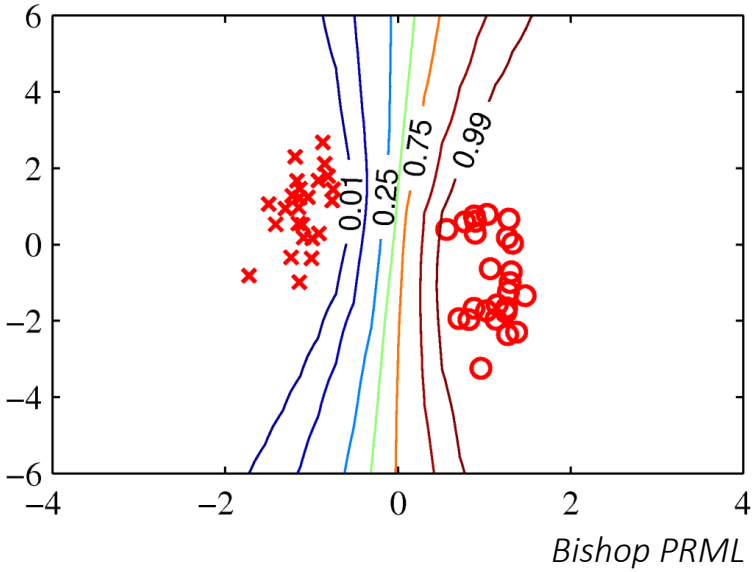
Bayesian Logistic Regression

Idea: In the separable case, there are *many perfect* linear classifiers which all separate the data. *Average* the classification result and accuracy using *all* of these classifiers.

- Optimal way to deal with missing knowledge in *Bayes sense*



Bishop PRML



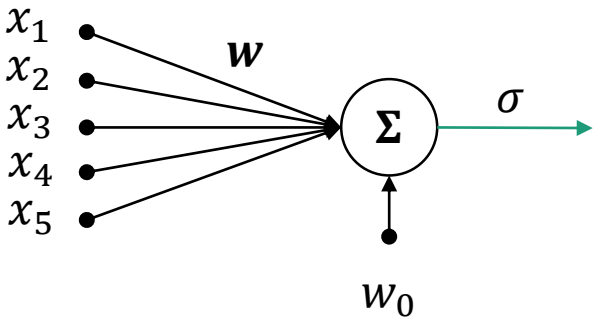
Bishop PRML

Logistic Regression and Neural Nets

- The standard single neuron with the logistic activation is logistic regression if trained with the same cost function (cross-entropy)

  But training with least-squares results in a different classifier

- Multiclass logistic regression with soft-max corresponds to what is called a *soft-max layer* in ANN. It is the standard multiclass output in most ANN architectures.



$$P(y = 1|\boldsymbol{x}, \boldsymbol{w}, w_0) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

Non-Linear Extension

- Logistic regression is often extended to non-linear cases: $\boldsymbol{x} \coloneqq \begin{bmatrix} \boldsymbol{x} \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$

  Extension through adding additional *transformed* features

  - Combination terms: $x_i x_j$
  - Monomial terms: x_i^2

  Standard procedure in medicine: inspect resulting $\boldsymbol{w}$ to find important factors and interactions $x_i x_j$ (comes with statistical information).

- Usage of *kernels* is possible: training and classification can be formulated with dot products of data points. The scalar products can be *"replaced"* by kernel expansions with the *kernel trick*.

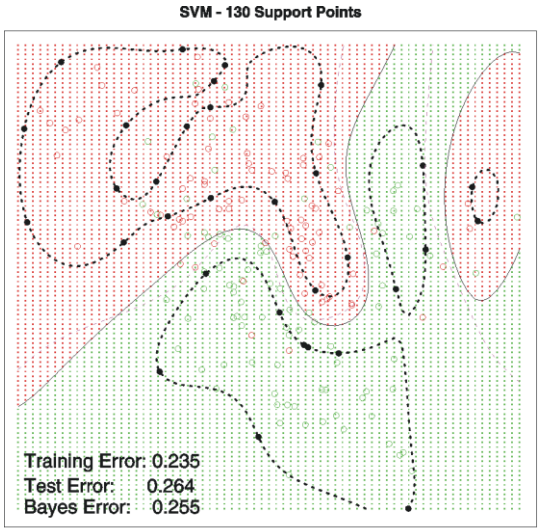
Kernel Logistic Regression

- Equations of logistic regression can be reformulated with dot products:

$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x} \rightarrow \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

- No Support Vectors: kernel evaluations with *all* training points

$$P(y = 1|\ \boldsymbol{x}) = \sigma\left(\sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})\right)$$



IVM (import vector machine):
Extension with only sparse support points

Ji Zhu & Trevor Hastie (2005) Kernel Logistic Regression and the Import Vector Machine, Journal of Computational and Graphical Statistics, 14:1, 185-205, DOI: 10.1198/106186005X25619

Discriminative vs. Generative

Comparison of logistic regression to naïve Bayes

> Ng, Andrew Y., and Michael I. Jordan. "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes." Advances in NIPS 14, 2001.

Conclusion:

- Logistic regression has a *lower* asymptotic error
- Naïve Bayes can reach its (higher) asymptotic error *faster*

General over-simplification (dangerous!): use a generative model with few data (more knowledge) and a discriminative model with a lot of training data (more learning)

Logistic Regression: Summary

- A probabilistic, linear method for classification!
- Discriminative method (Model for posterior)
- Linear model for the Logit

$$\log \frac{p}{1-p} = \langle \tilde{w}, \tilde{x} \rangle$$

- The posterior probability is given by the logistic function of the Logit:

$$P(y=1|x) = \sigma(\langle \tilde{w}, \tilde{x} \rangle) = \frac{1}{1+\exp(-\langle \tilde{w}, \tilde{x} \rangle)}$$

- ML-estimation of $\tilde{w}$ is unique but non-linear
- Logistic regression is a very often used method
- Extendable to multiclass
- General Purpose method, included in every standard software, e.g. `glm` in R, `glmfit/glmval` in Matlab – its easy to apply!