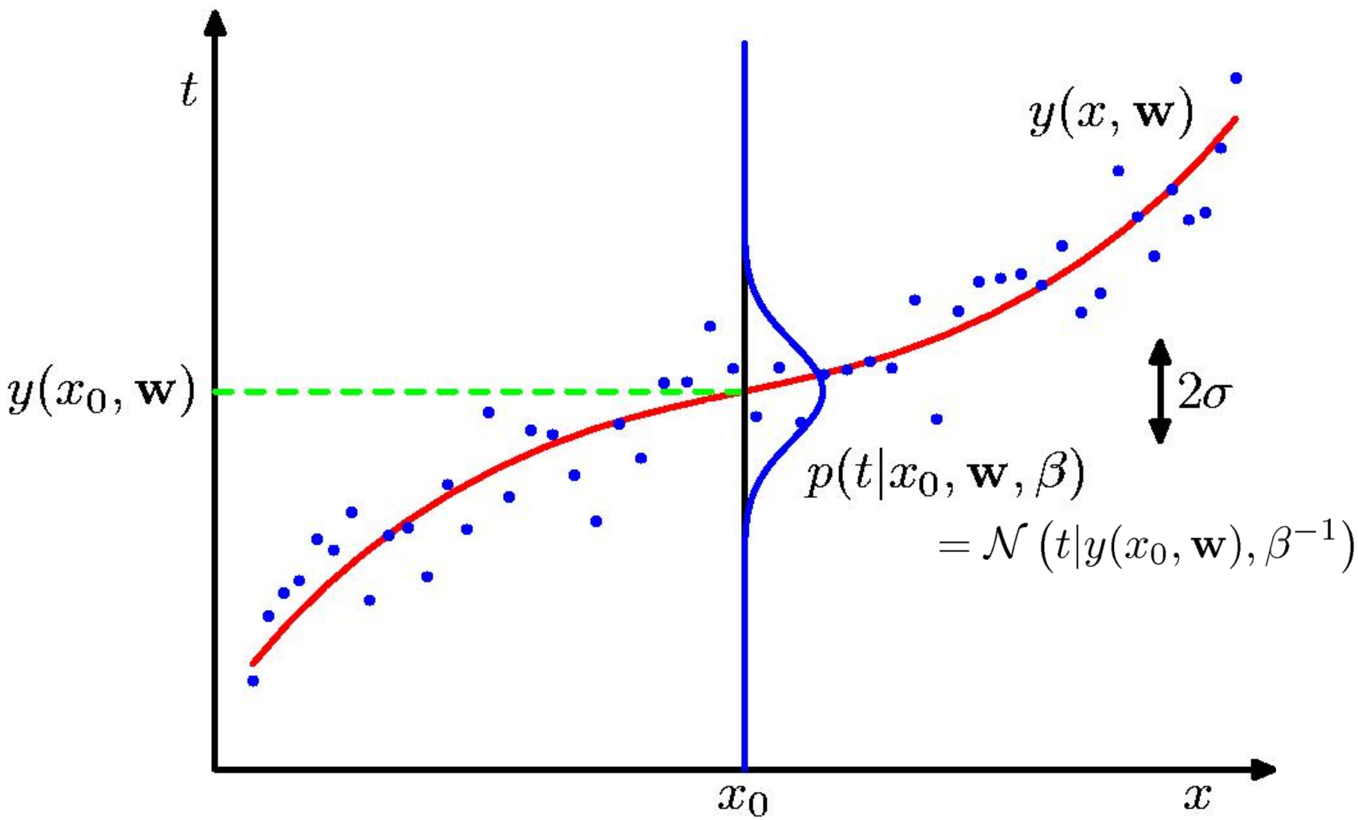
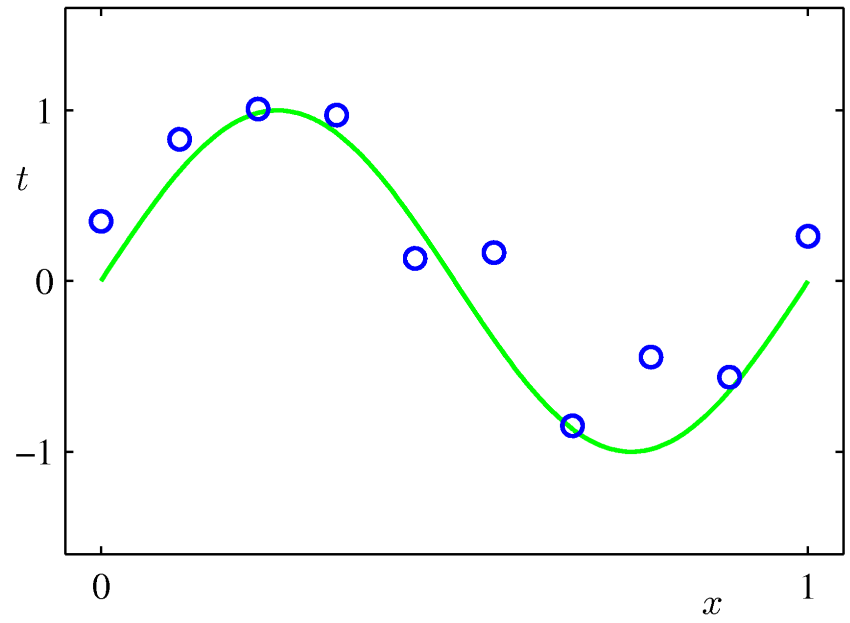
Curve Fitting Re-visited



Linear Basis Function Models (1)

Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Linear Basis Function Models (2)

Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

Where $\phi_j(x)$ are known as *basis functions*.

Typically, $\phi_0(x) = 1$, so that w_0 acts as a bias.

In the simplest case, we use linear basis functions : $\phi_d(x) = x_d$.

Maximum Likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|y(x_n, \mathbf{w}), \beta^{-1}\right)$$

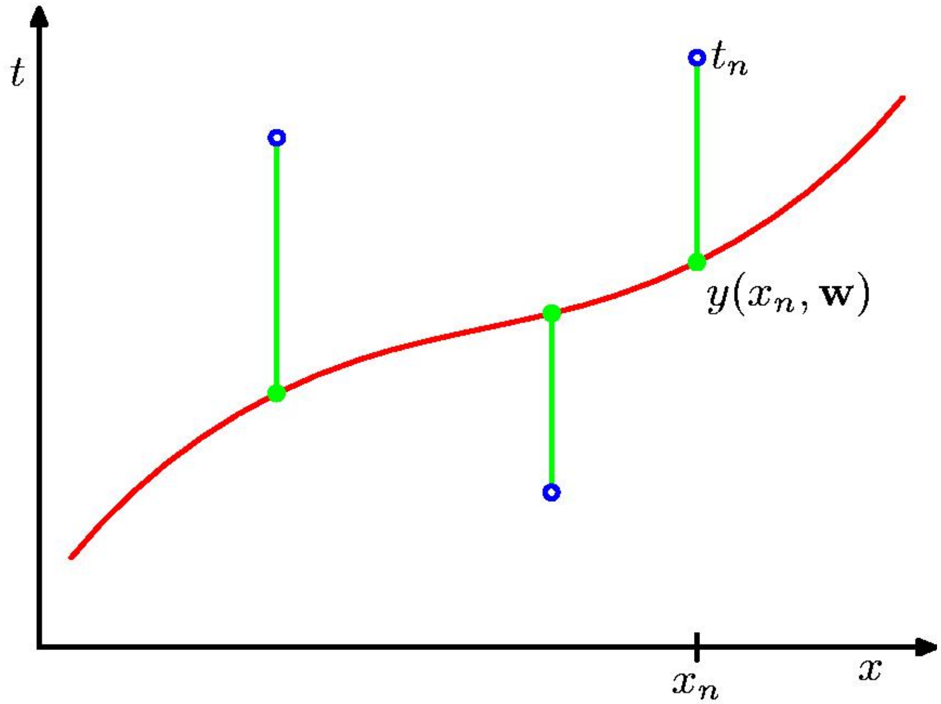
Data $\mathbf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$
$\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \underbrace{-\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2}_{\beta E(\mathbf{w})} + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

Determine $\mathbf{w}_{\mathrm{ML}}$ by minimizing sum-of-squares error, $E(\mathbf{w})$.
Determine also the precision parameter (inverse variance):

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2$$

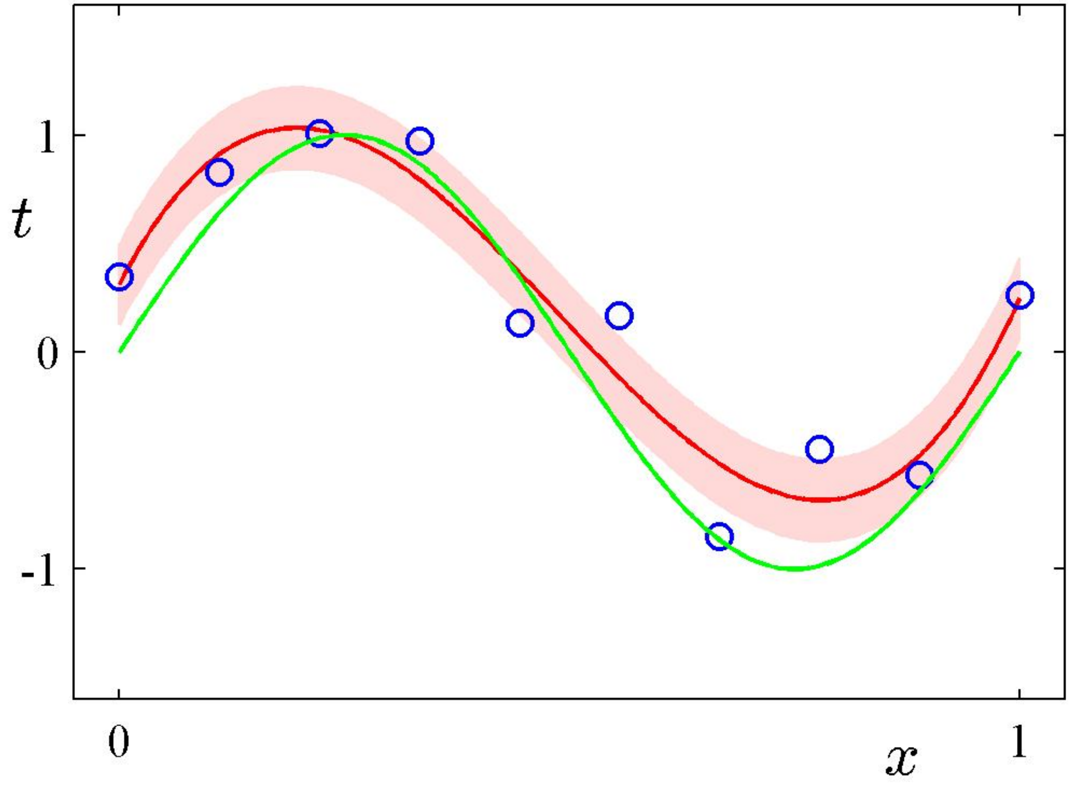
Sum-of-Squares Error Function



$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

Predictive Distribution

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$



MAP: A Step towards Bayes

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

$$\beta\widetilde{E}(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Determine $\mathbf{w}_{\mathrm{MAP}}$ by minimizing regularized sum-of-squares error, $\widetilde{E}(\mathbf{w})$.

Bayesian Curve Fitting

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w}) p(\mathbf{w}|\mathbf{x}, \mathbf{t}) \, \mathrm{d}\mathbf{w} = \mathcal{N}\left(t|m(x), s^2(x)\right)$$

Training data

$$\mathbf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$$
$$\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$$

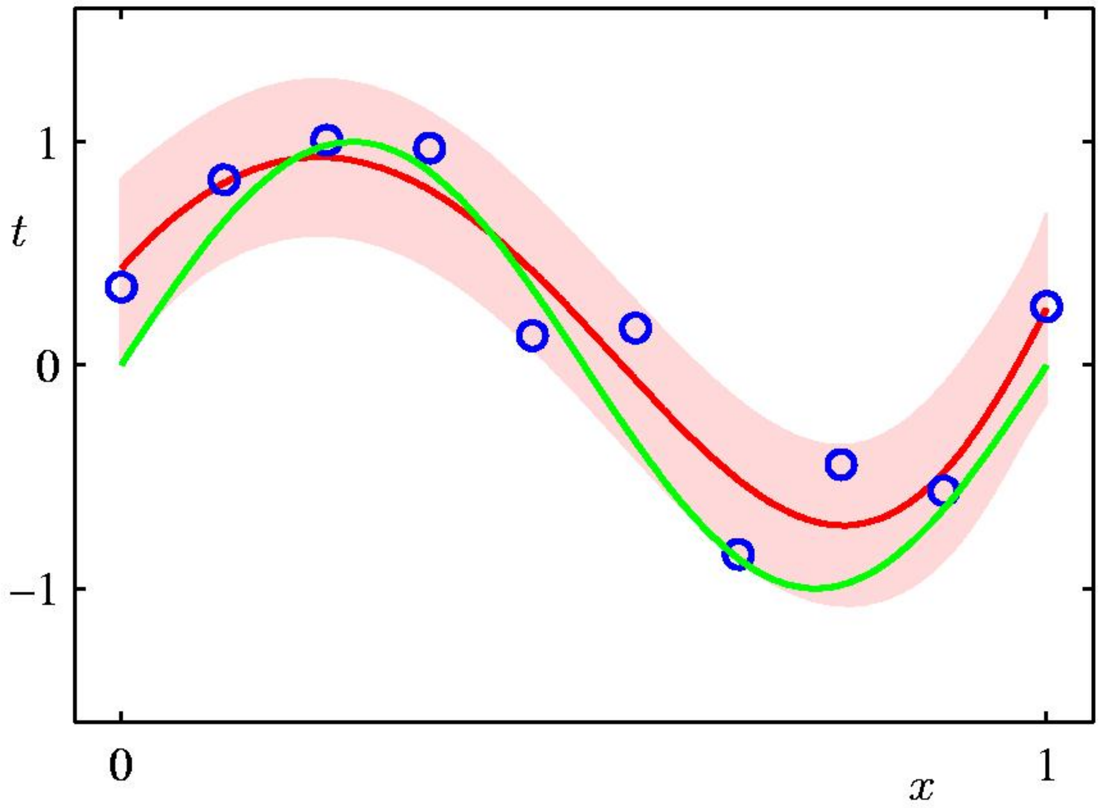
$$m(x) = \beta \boldsymbol{\phi}(x)^{\mathrm{T}} \mathbf{S} \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n \qquad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}} \mathbf{S} \boldsymbol{\phi}(x)$$

Where $\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^{\mathrm{T}}$

E.g. polynomials as basis functions $\boldsymbol{\phi}(x_n) = \left(x_n^0, \ldots, x_n^M\right)^{\mathrm{T}}$

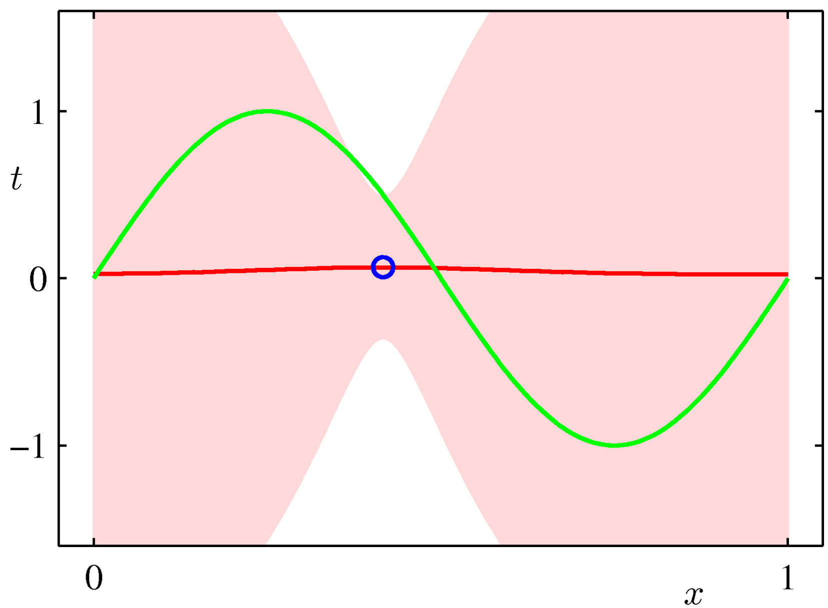
Bayesian Predictive Distribution

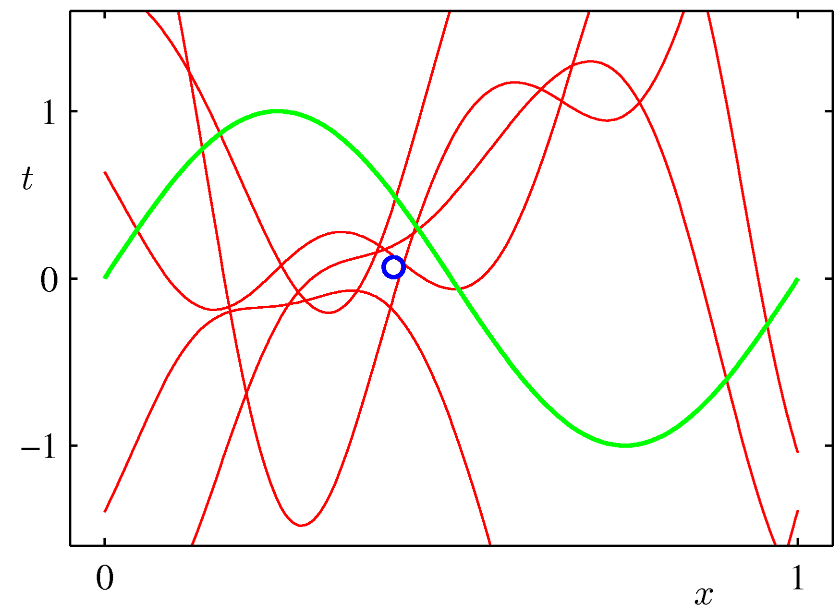
$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$



Predictive Distribution (2)

Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point



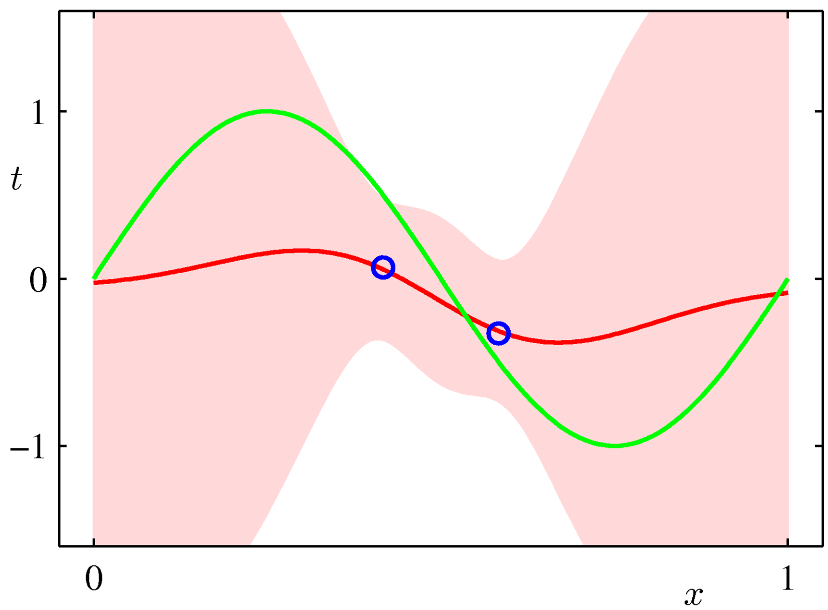


$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$

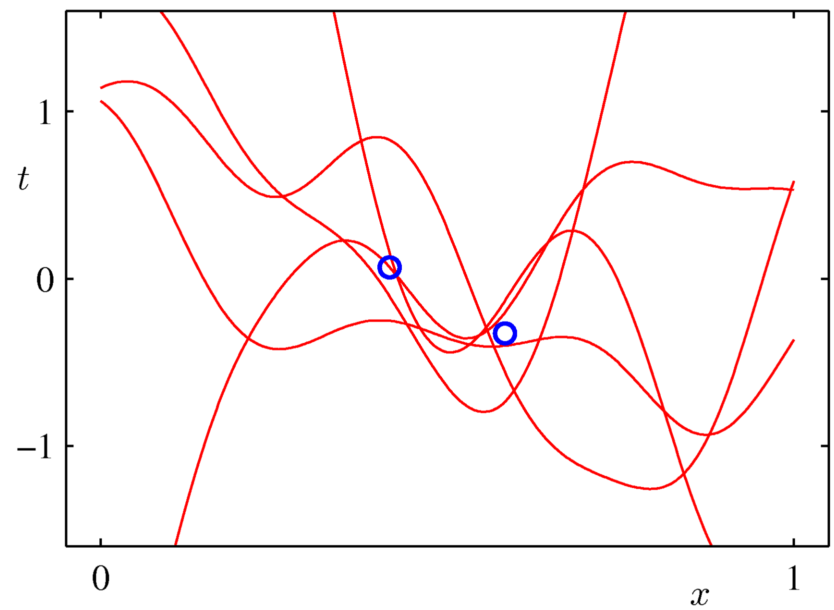
$$y(x, \mathbf{w})$$

Predictive Distribution (3)

Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



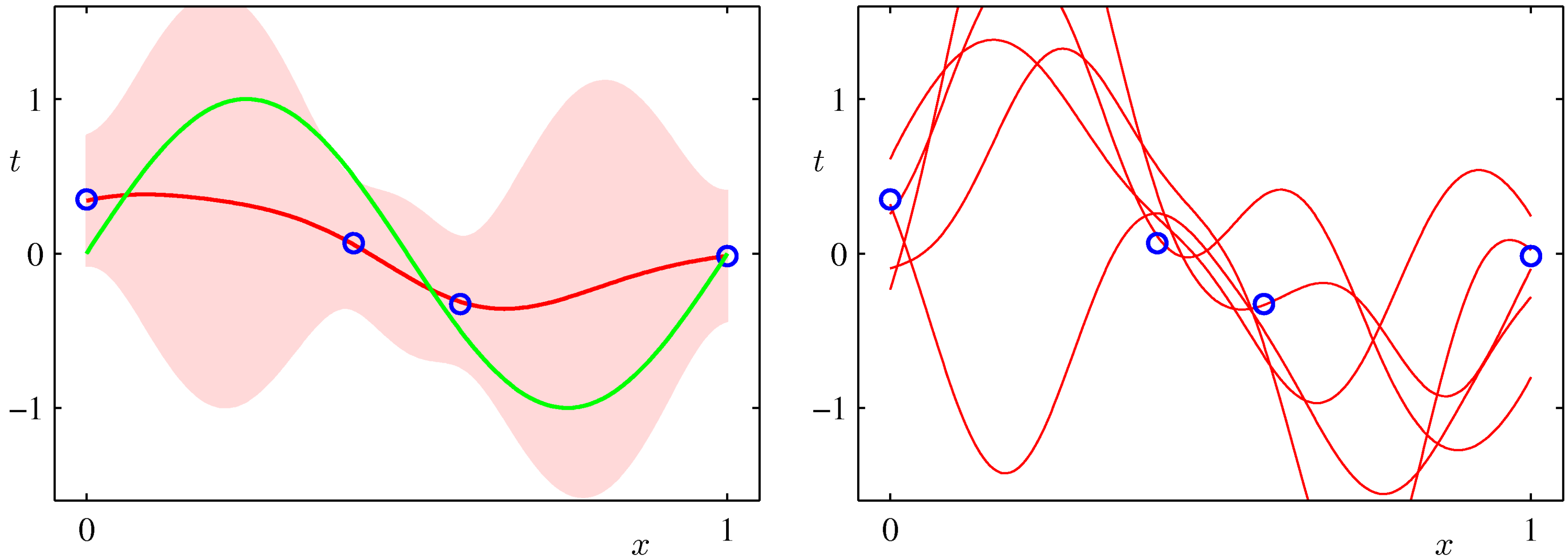
$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$



$$y(x, \mathbf{w})$$

Predictive Distribution (4)

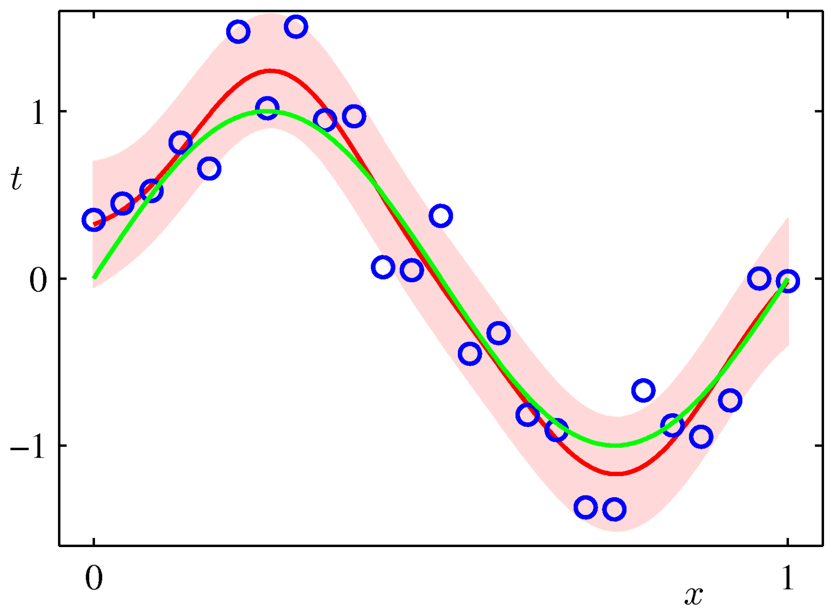
Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



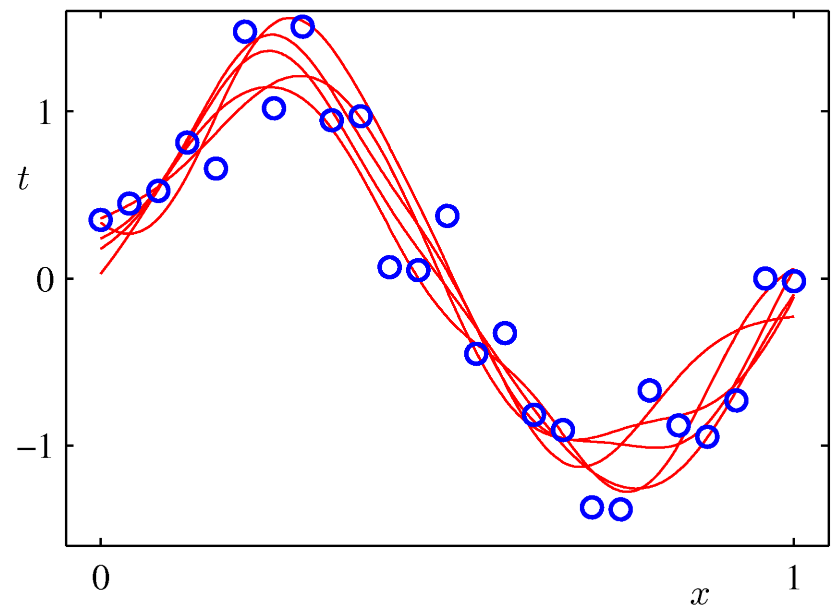
$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right) \qquad y\left(x, \mathbf{w}\right)$$

Predictive Distribution (5)

Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points



$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$



$$y\left(x, \mathbf{w}\right)$$

Regression vs. Classification

Regression:

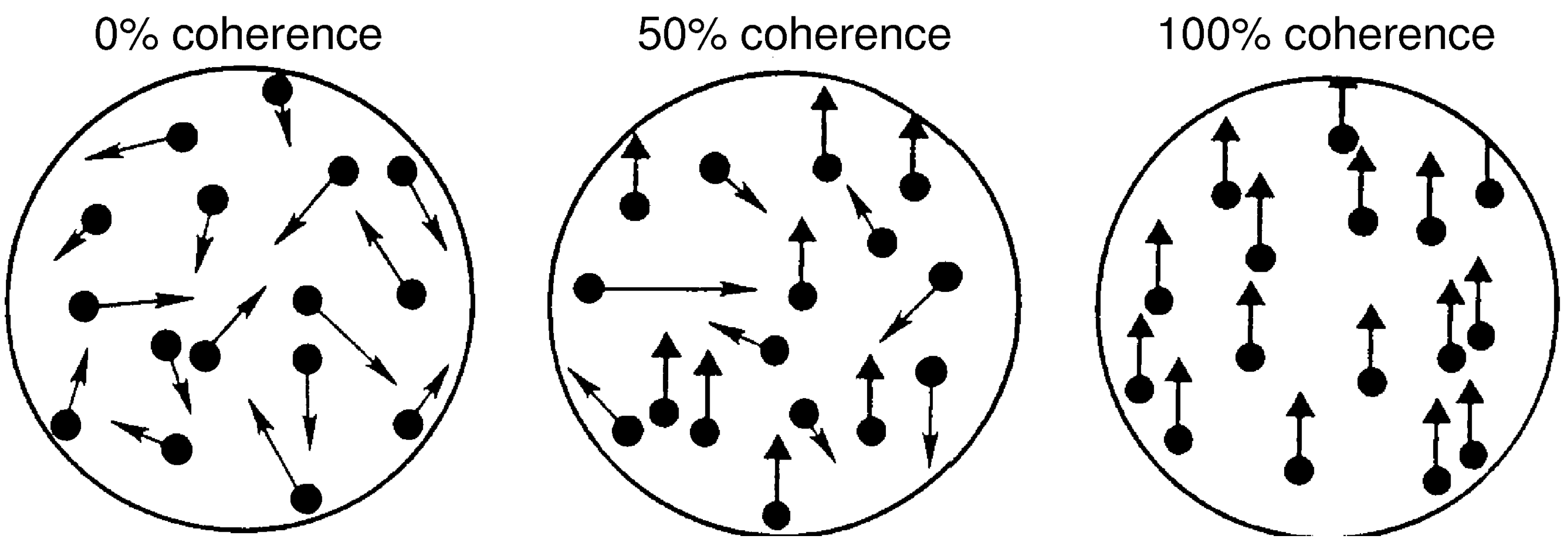
$$x \in [-\infty, \infty], t \in [-\infty, \infty]$$

Classification:

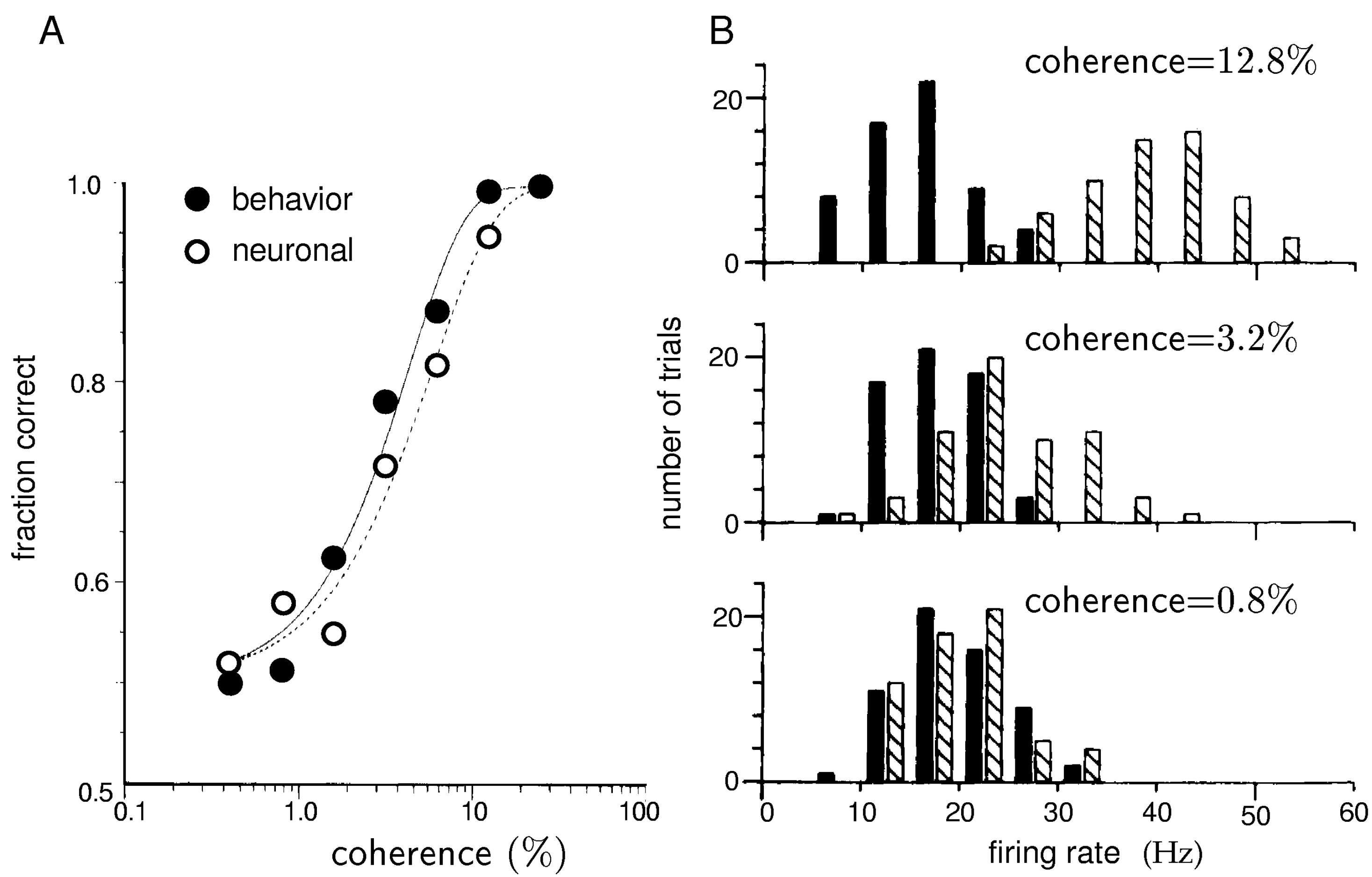
$$x \in [-\infty, \infty], t \in \{0, 1\}$$

Neural Example: neuron in MT

- Middle temporal cortex: large receptive fields sensitive to object motion
- record from single neuron during movement patterns such as the ones below
- animal is trained to decide if the coherent movement is upwards or downwards



- Left: behavioral performance of the animal and of an “ideal observer” considering single neuron
- Right: histograms (thinned) of average firing rate for different stimuli (up/down) at different coherence levels



Maximum likelihood

Optimal strategy for discriminating between two alternative signals presented in background of noise?

Let's call the two alternative signals: + and –

Assume we must base our decisions on the observation of a single observable x
x could be e.g. the firing rate of a neuron when x is present

If the signal is + then the values of x are chosen from $P(x|+)$
If the signal is - then the values of x are chosen from $P(x|-)$

If we have seen a particular value of x, can we tell which signal was presented?

Intuition: Divide x axis at critical point x_0: Everything to right is called a +, everything to the left a -.

How should we choose x_0 ?

Maximum likelihood

Compute probability of correct decision as function of threshold…
…then find the value of the threshold that maximizes this probability!

Probability of correctly identifying signal +:

$$P(\text{say } + | \text{signal is } +) = \int_{x_0}^{\infty} dx P(x|+)$$

Probability of correctly identifying signal -:

$$P(\text{say -} | \text{signal is -}) = \int_{\infty}^{x_0} dx P(x|-)$$

Probability of making correct choice:

$$P_c(x_0) = P(+) \int_{x_0}^{\infty} dx P(x|+) + P(-) \int_{\infty}^{x_0} dx P(x|-)$$

Maximum likelihood

Probability of making correct choice:

$$P_c(x_0) = P(+)\int_{x_0}^{\infty} dx P(x|+) + P(-)\int_{\infty}^{x_0} dx P(x|-)$$

Maximize it!

$$\frac{dP_c(x_0)}{dx_0} = 0$$

$$P(+)\frac{d}{dx_0}\int_{x_0}^{\infty} dx P(x|+) + P(-)\frac{d}{dx_0}\int_{\infty}^{x_0} dx P(x|-) = 0$$

$$\Rightarrow \quad -P(+)\ P(x_0|+) + P(-)P(x_0|-) = 0$$

$$\Rightarrow \quad P(+)\ P(x_0|+) = P(-)P(x_0|-)$$

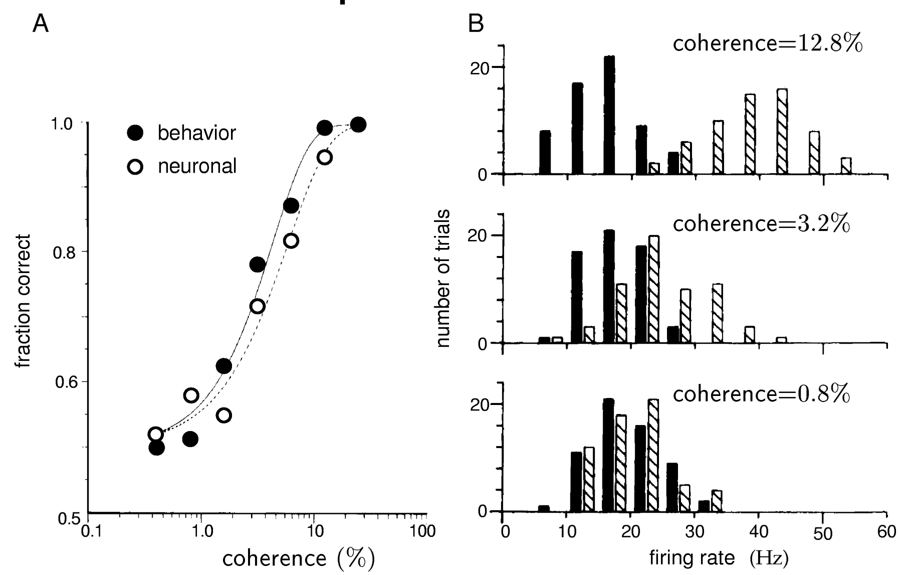
Maximum likelihood

$$P(+)P(x_0|+) = P(-)P(x_0|-)$$

In the simple case that signals x and – are equally likely, i.e. P(+)=P(-)

$$\Rightarrow \quad P(x_0|+) = P(x_0|-)$$

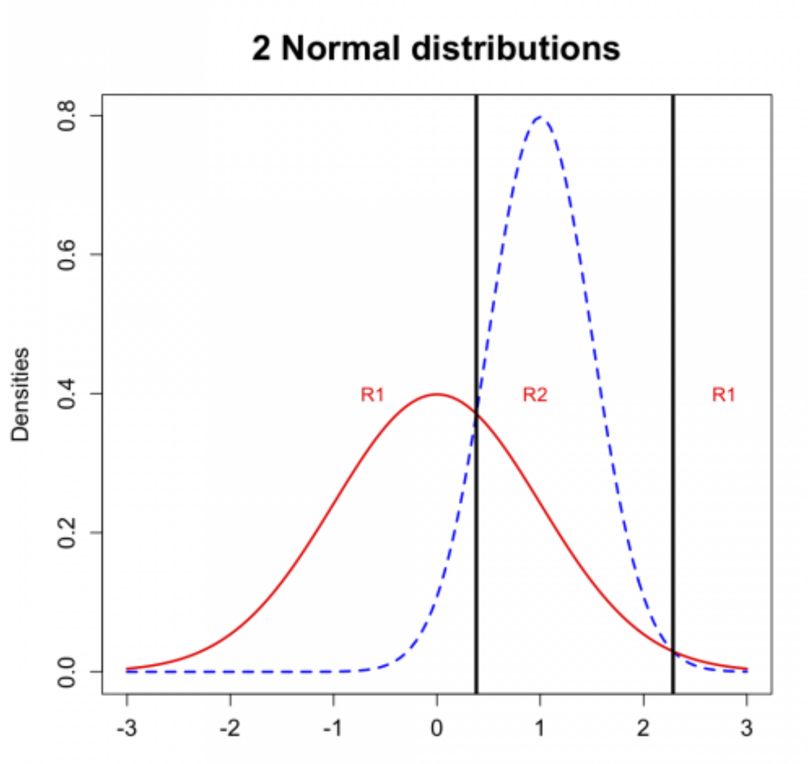
Set threshold where two probabilities cross



Maximum likelihood

$$\Rightarrow \quad P(x_0|+) = P(x_0|-)$$

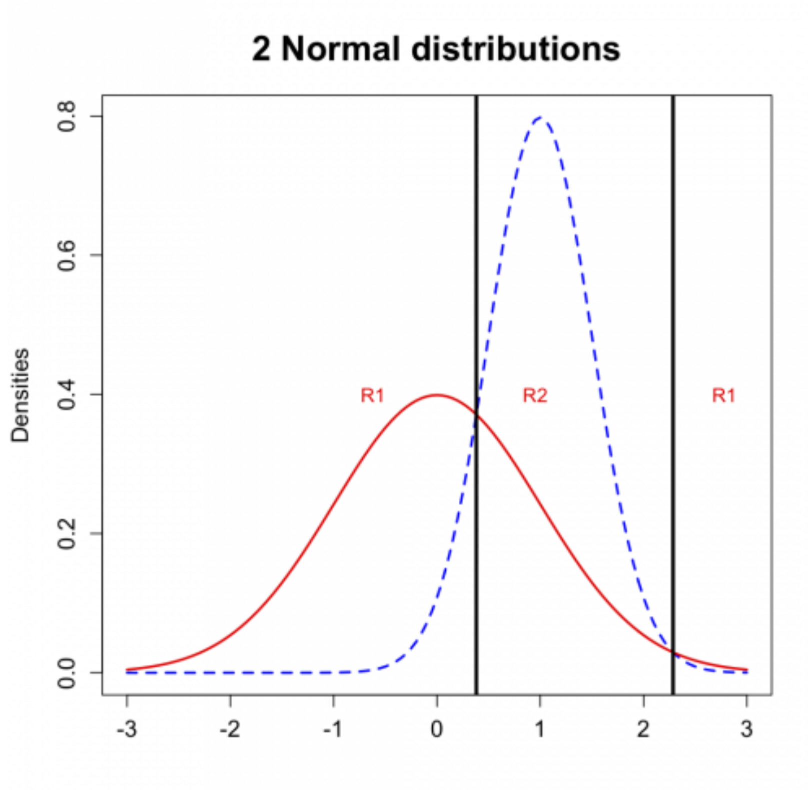
There can be several dividing lines



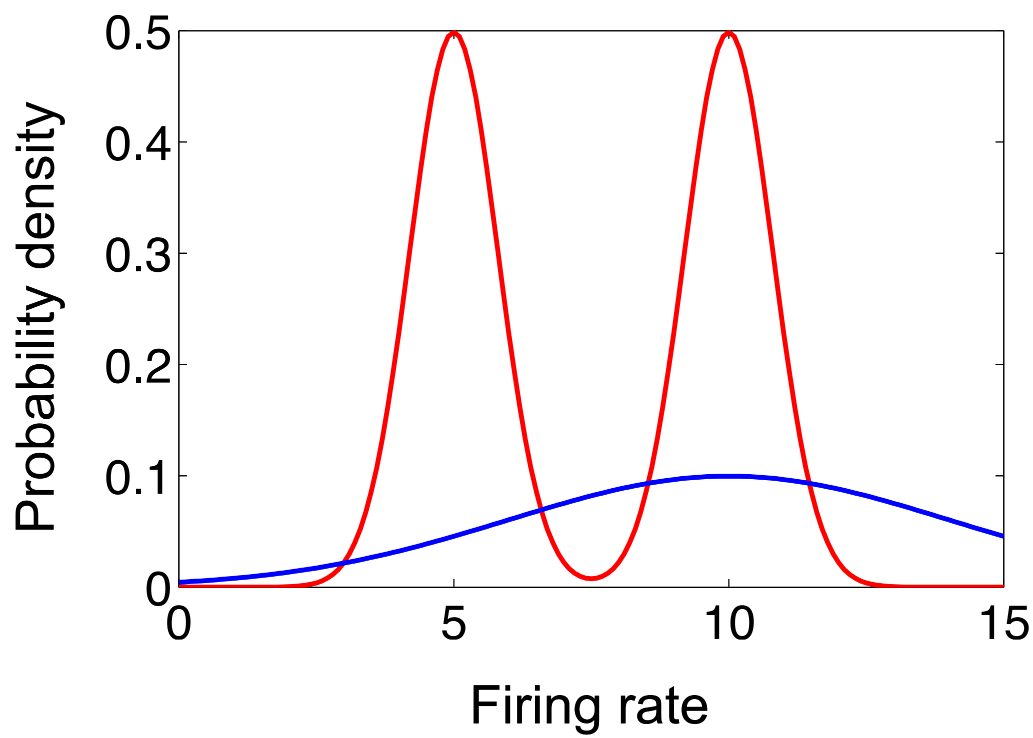
Maximum likelihood

In general: One cannot do better than the likelihood ratio

$$l(x) = \frac{P(x|+)}{P(x|-)} = \frac{L(+|x)}{L(-|x)}$$

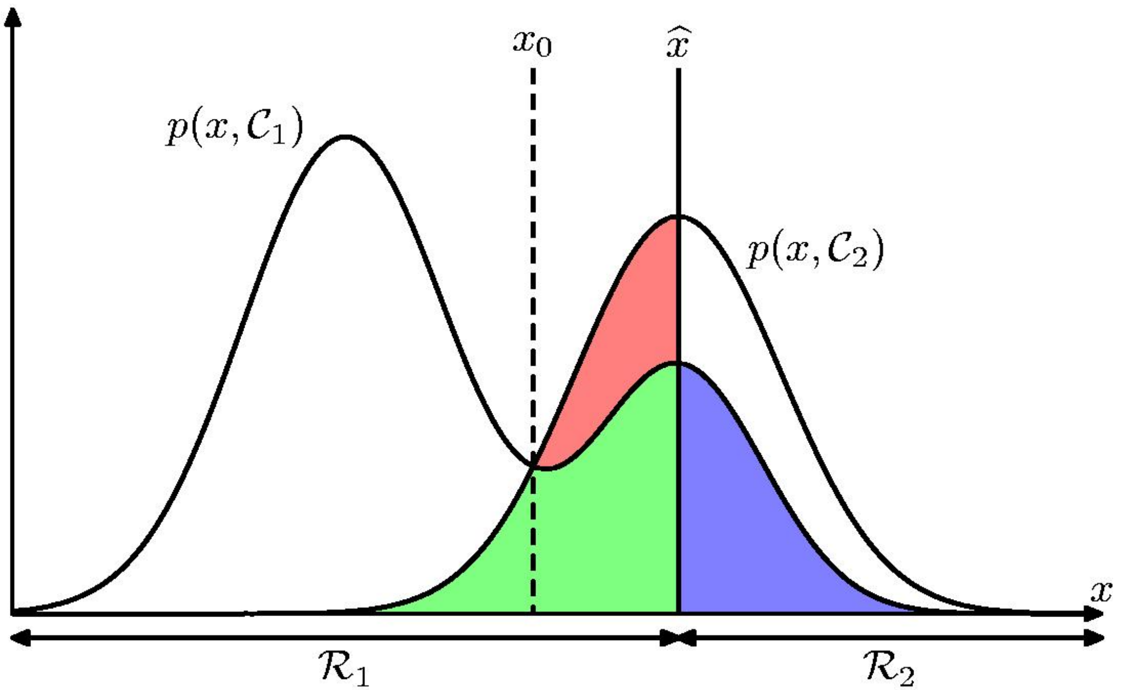


Very general result. Applies also to multimodal and multivariate distributions.



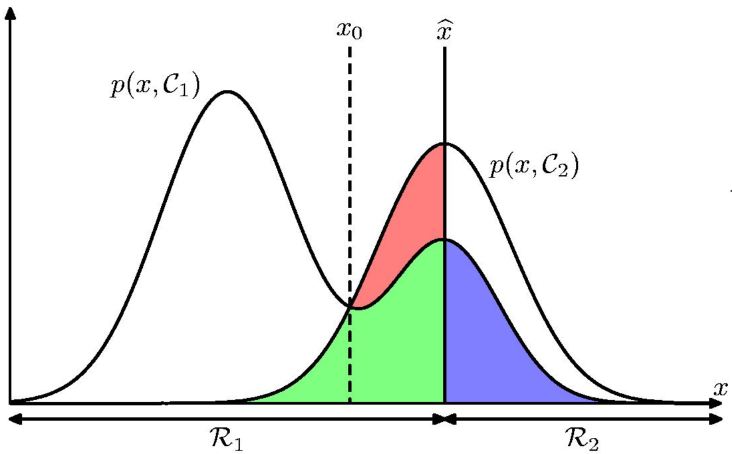
Alternative method: likelihood ratio $$\frac{L(\circ|x)}{L(\circ|x)} = \frac{p(x|\circ)}{p(x|\circ)}$$

Minimum Misclassification Rate



$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}. \end{aligned}$$

Minimum Misclassification Rate



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}.
\end{aligned}
$$

We are free to choose the decision rule that assigns each point x to one of the two classes.

To minimize integrand: $p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$ must be small

Assign x to class for which the posterior $p(\mathcal{C}_k|\mathbf{x})$ is larger!

Three strategies

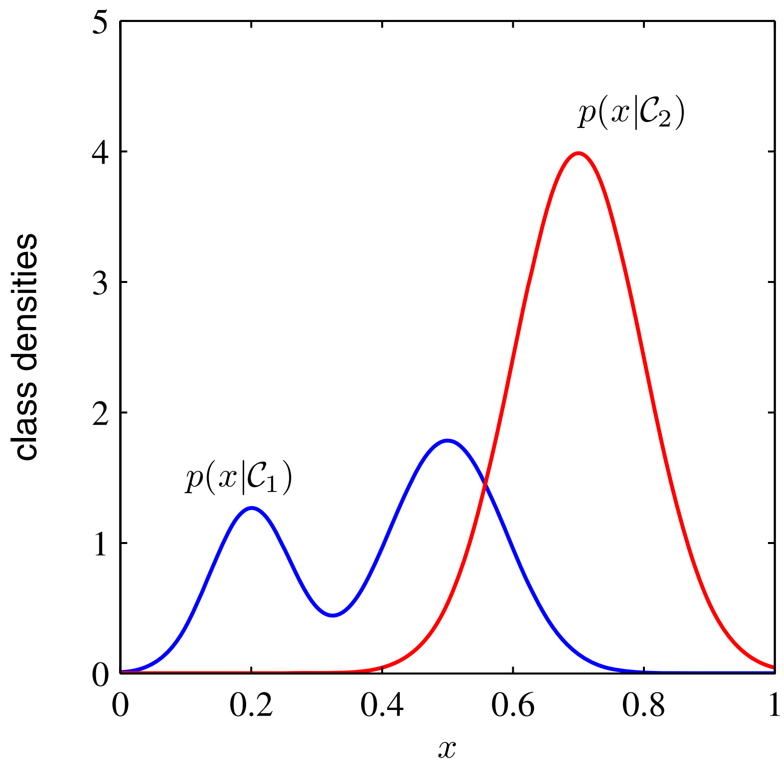
1. Modeling the class-conditional density for each class C_k, and prior, then use Bayes

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

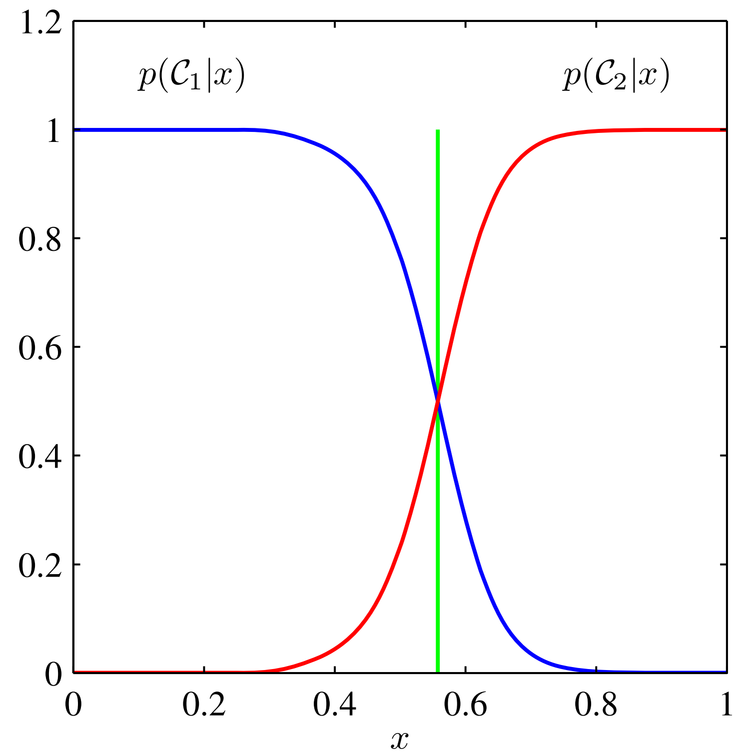
2. First solve the inference problem of determining the posterior class probabilities $p(C_k|x)$, and then subsequently use decision theory to assign each new x to one of the classes
3. Find discriminant function that directly maps x to class label

Class-conditional density vs. posterior

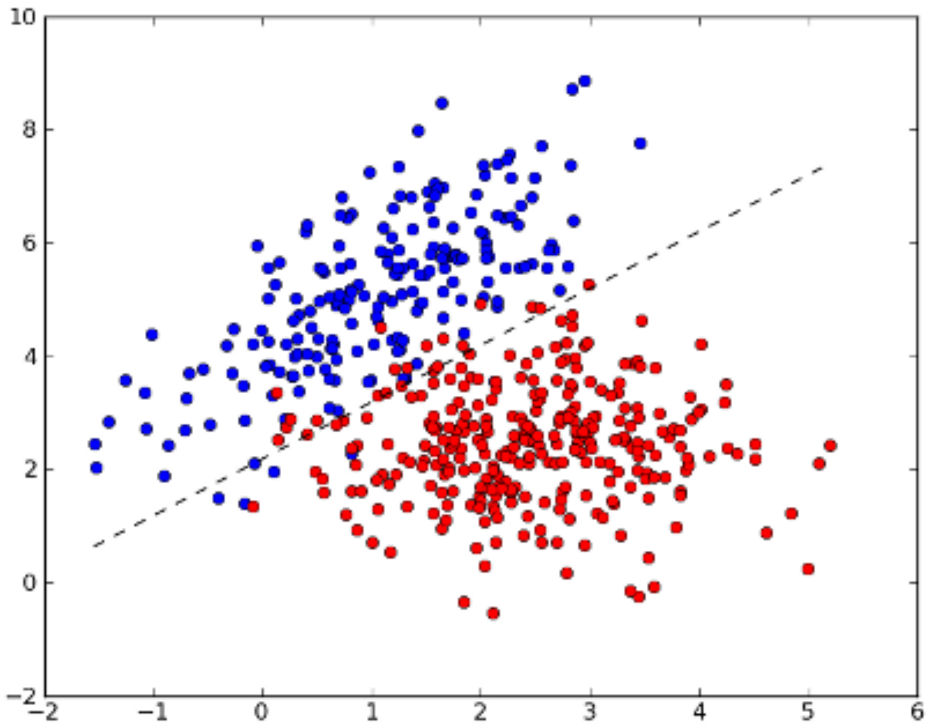
Class-conditional densities



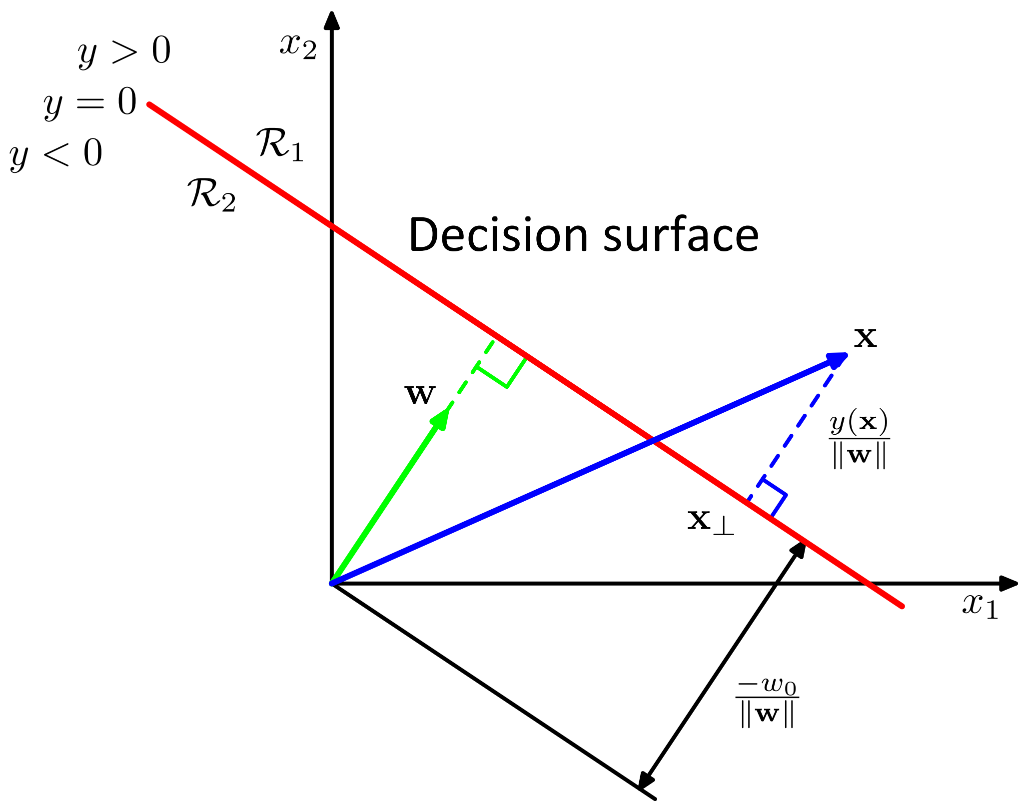
Posterior probabilities



Several dimensions



Several dimensions



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

weight vector

bias

$\mathcal{C}_1$ if $y(\mathbf{x}) \geqslant 0$

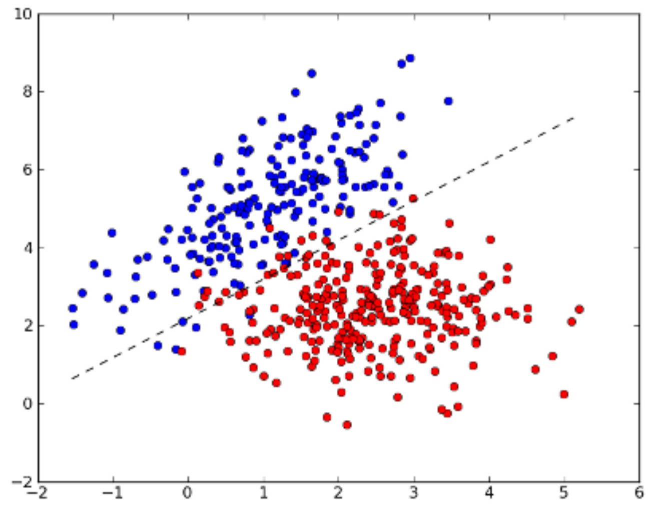
$\mathcal{C}_2$ otherwise

Fisher's linear discriminant 1

Projecting data down to one dimension

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

But how?



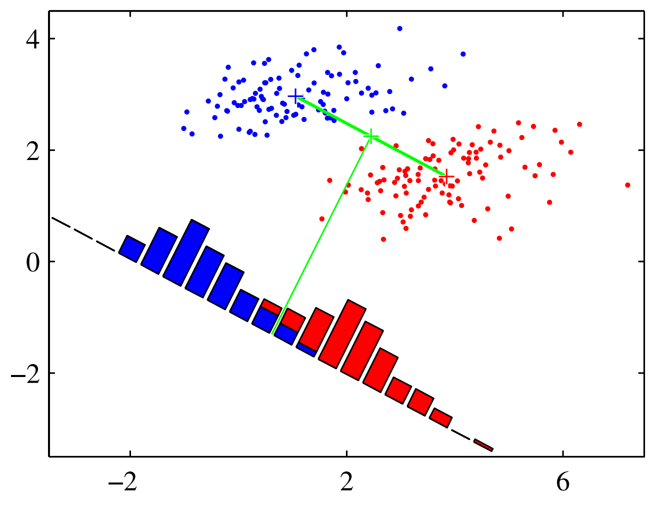
Fisher's linear discriminant 2

Define class means

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$

Try maximize

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1)$$



Fisher's linear discriminant 3

Instead, consider: ratio of between class variance to within class variance

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

With

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

Called Fisher criterion. Maximize it!

Fisher's linear discriminant 4

Maximizing the Fisher Criterion we obtain

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

with the total within class covariance

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}$$

This is called Fisher's linear discriminant

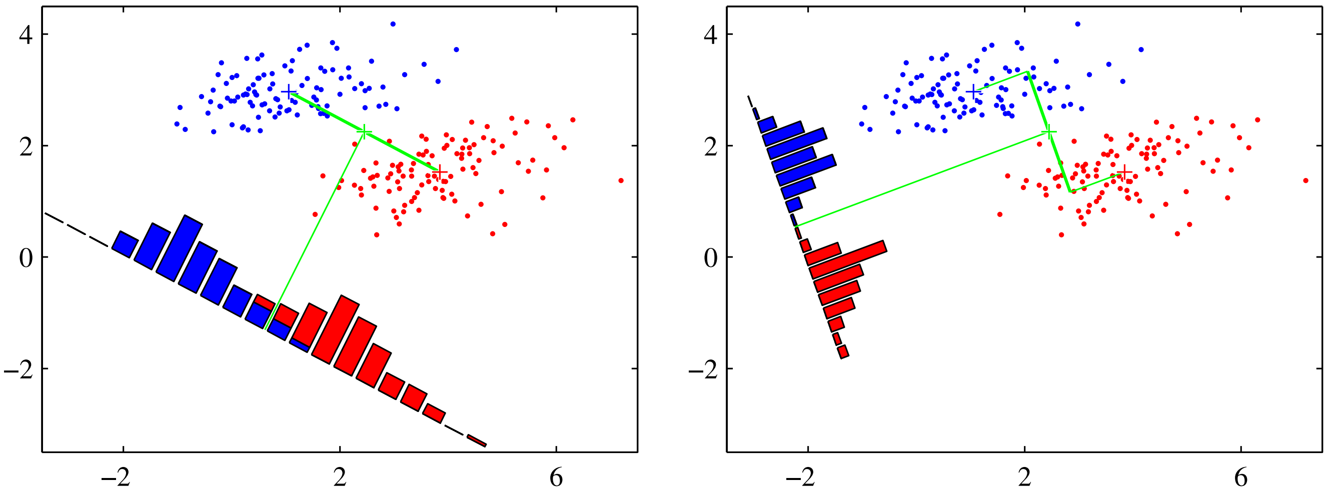
Fisher’s linear discriminant 4

Fisher’s linear discriminant

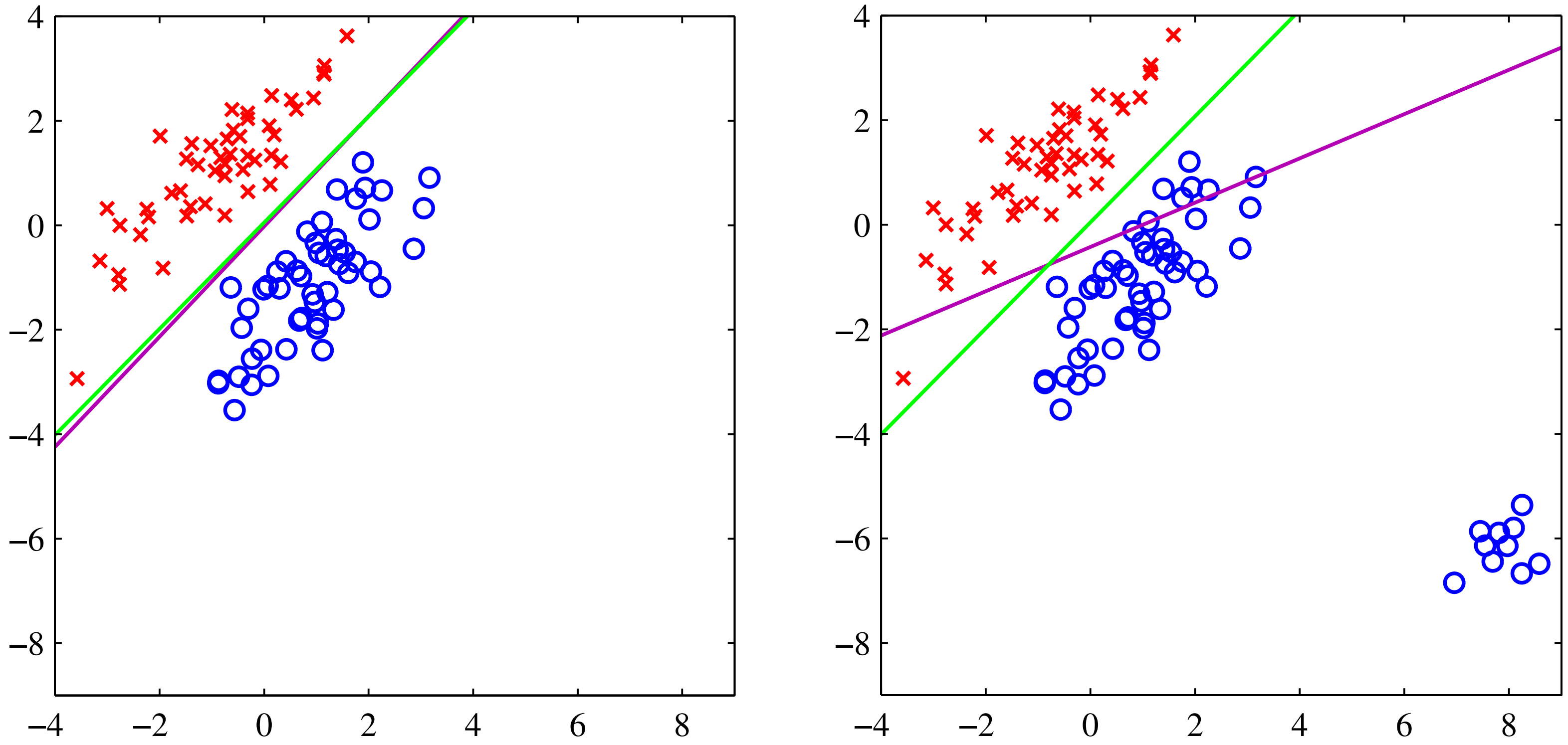
$$\mathbf{w} \propto \mathbf{S}_\mathrm{W}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Fisher Criterion

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

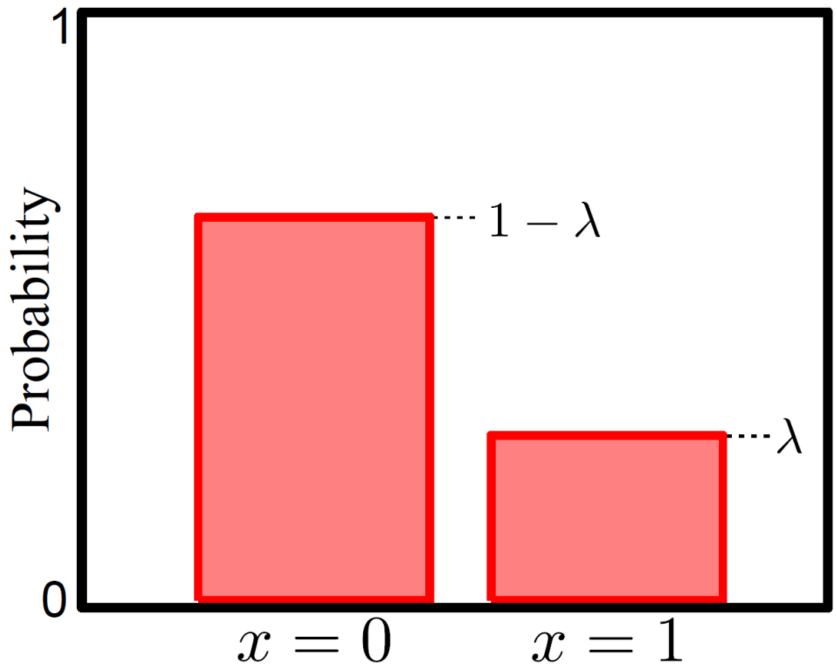


Least squares for classification fails



Use logistic regression instead!

Bernoulli Distribution



$$Pr(x=0) = 1-\lambda$$
$$Pr(x=1) = \lambda.$$

or

$$Pr(x) = \lambda^x(1-\lambda)^{1-x}$$

For short we write:

$$Pr(x) = \text{Bern}_x[\lambda]$$

Bernoulli distribution describes situation where only two possible outcomes y=0/y=1 or failure/success

Takes a single parameter $\lambda \in [0,1]$

Logistic Regression

Consider two class problem.

- Choose Bernoulli distribution over world.
- Make parameter λ a function of x

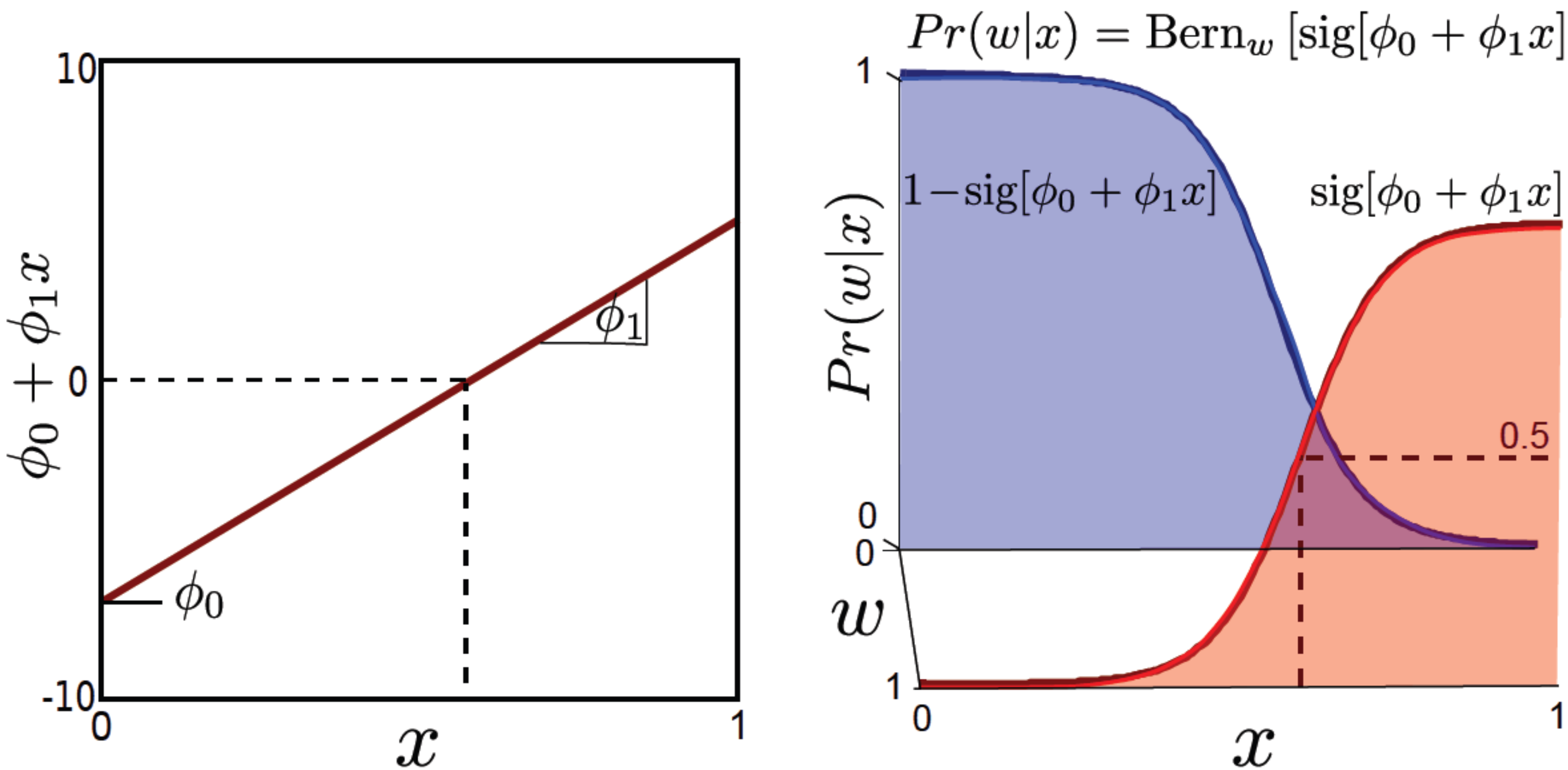
$$Pr(w|\phi_0, \boldsymbol{\phi}, \mathbf{x}) = \text{Bern}_w[\text{sig}[a]]$$

Model activation with a linear function

$$a = \phi_0 + \boldsymbol{\phi}^T \mathbf{x}$$

creates number between $[-\infty, \infty]$. Maps to $[0, 1]$ with

$$\text{sig}[a] = \frac{1}{1 + \exp[-a]}$$



Two parameters

$$\boldsymbol{\theta} = \{\phi_0, \phi_1\}$$

Learning by standard methods (ML,MAP, Bayesian)
Inference: Just evaluate Pr(w|x)

Neater Notation

$$Pr(w|\phi_0, \boldsymbol{\phi}, \mathbf{x}) \quad = \quad \text{Bern}_w\left[\text{sig}[a]\right]$$

To make notation easier to handle, we

- Attach a 1 to the start of every data vector

$$\mathbf{x}_i \leftarrow [1 \quad \mathbf{x}_i^T]^T$$

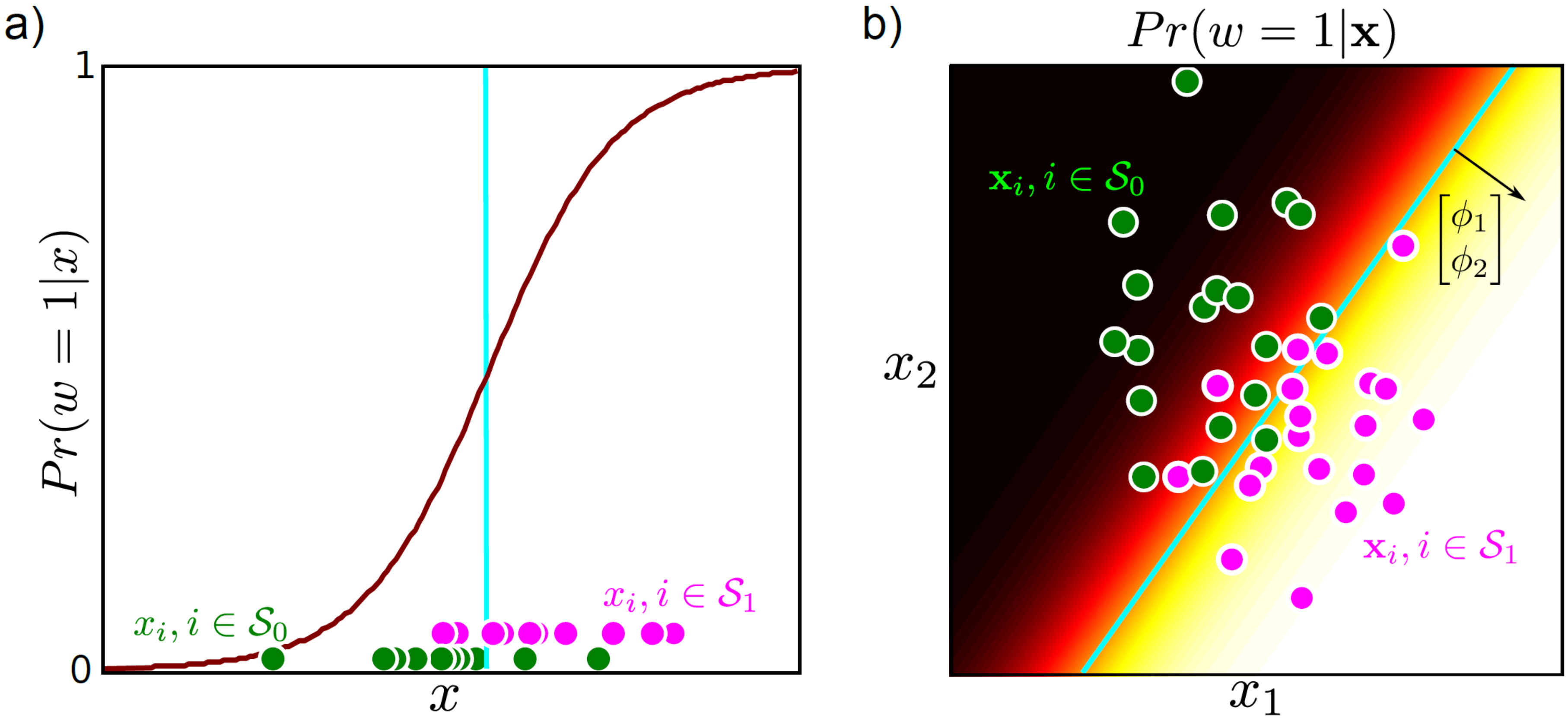
- Attach the offset to the start of the gradient vector $\boldsymbol{\phi}$

$$\boldsymbol{\phi} \leftarrow [\phi_0 \quad \boldsymbol{\phi}^T]^T$$

New model:

$$Pr(w|\boldsymbol{\phi}, \mathbf{x}) = \text{Bern}_w\left[\frac{1}{1 + \exp[-\boldsymbol{\phi}^T\mathbf{x}]}\right]$$

Logistic regression



$$Pr(w|\boldsymbol{\phi}, \mathbf{x}) = \text{Bern}_w\left[\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}]}\right]$$

Maximum Likelihood

$$Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\phi}) = \prod_{i=1}^{I} \lambda^{w_i}(1-\lambda)^{1-w_i}$$

$$= \prod_{i=1}^{I} \left(\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right)^{w_i} \left(\frac{\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right)^{1-w_i}$$

Take logarithm

$$L = \sum_{i=1}^{I} w_i \log\left[\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right] + \sum_{i=1}^{I}(1-w_i)\log\left[\frac{\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right]$$

Take derivative:

$$\frac{\partial L}{\partial \boldsymbol{\phi}} = -\sum_{i=1}^{I}\left(\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]} - w_i\right)\mathbf{x}_i = -\sum_{i=1}^{I}(\text{sig}[a_i] - w_i)\,\mathbf{x}_i$$

Derivatives

$$\frac{\partial L}{\partial \boldsymbol{\phi}} = -\sum_{i=1}^{I}\left(\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]} - w_i\right)\mathbf{x}_i = -\sum_{i=1}^{I}\left(\text{sig}[a_i] - w_i\right)\mathbf{x}_i$$

Unfortunately, there is no closed form solution– we cannot get an expression for ϕ in terms of x and w

Have to use a general purpose technique:

“iterative non-linear optimization”

Optimization

Goal:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[f[\boldsymbol{\theta}] \right]$$

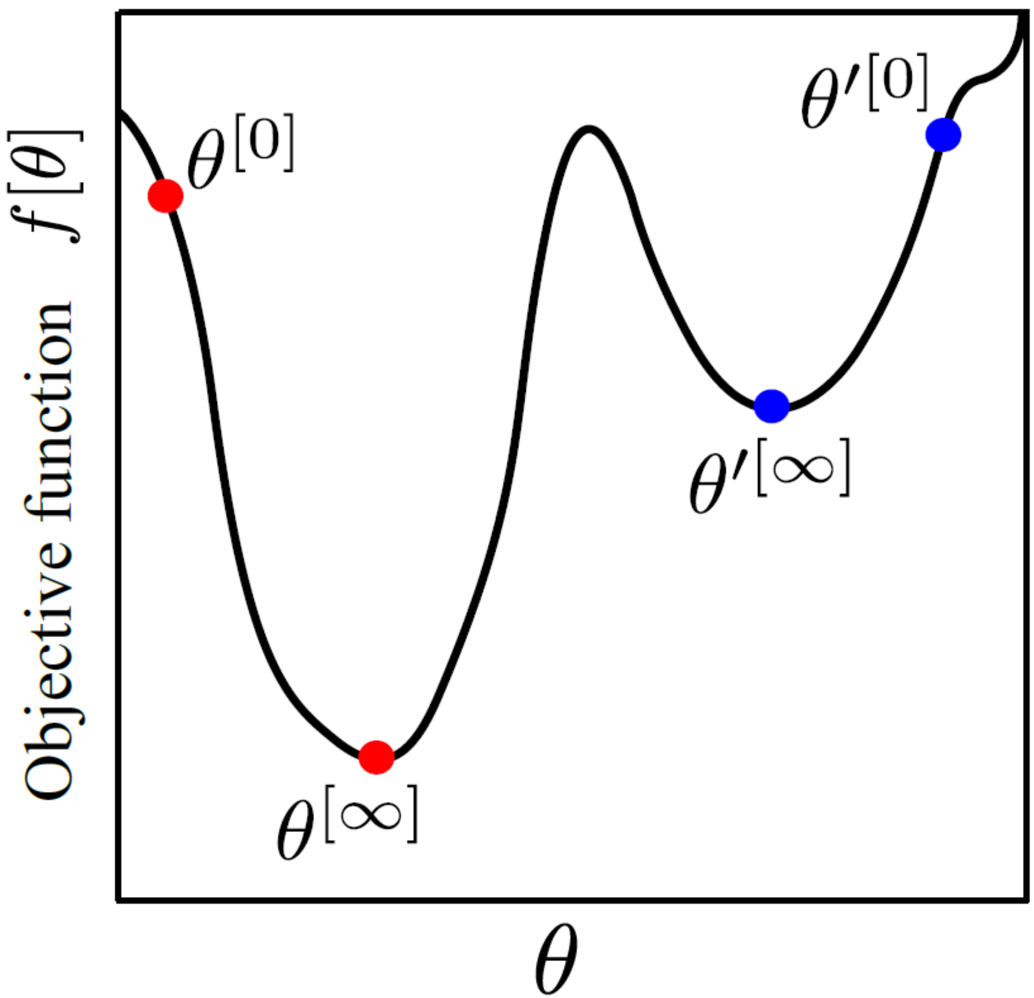
Cost function or Objective function

How can we find the minimum?

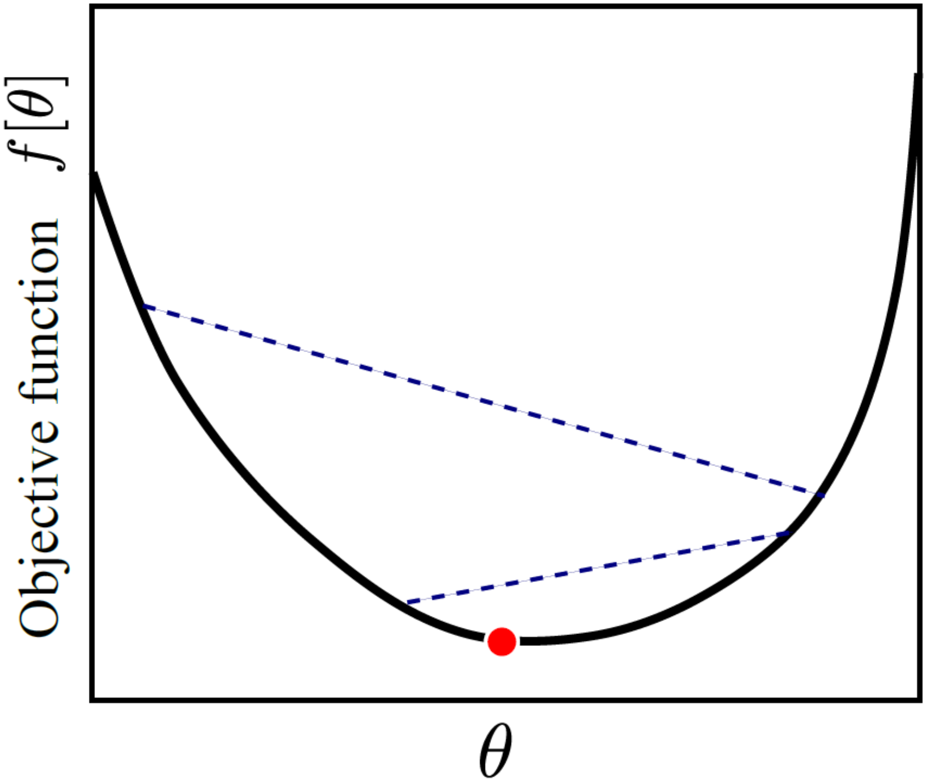
Basic idea:

- Start with estimate $\boldsymbol{\theta}^{[0]}$
- Take a series of small steps to $\boldsymbol{\theta}^{[1]}, \boldsymbol{\theta}^{[2]} \ldots \boldsymbol{\theta}^{[\infty]}$
- Make sure that each step decreases cost
- When can't improve, then must be at minimum

Local Minima

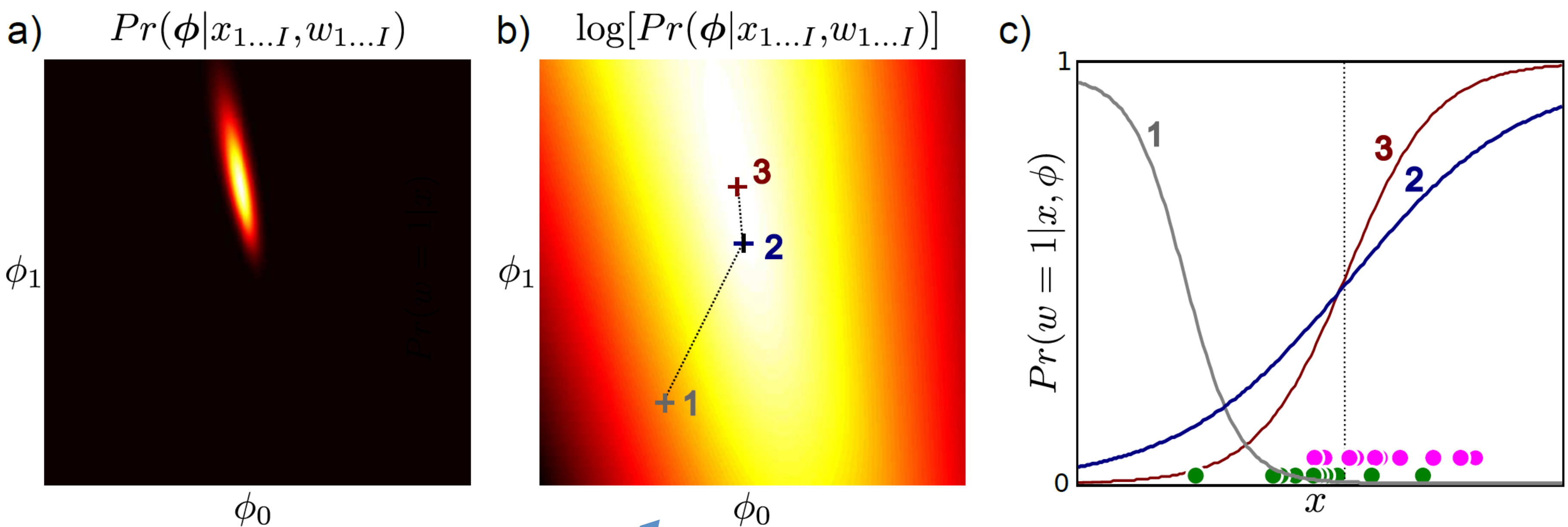


Convexity



If a function is convex, then it has only a single minimum.
Can tell if a function is convex by looking at 2nd derivatives

$$Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\phi}) = \prod_{i=1}^{I}\left(\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right)^{w_i}\left(\frac{\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right)^{1-w_i}$$



$$L = \sum_{i=1}^{I} w_i \log\left[\frac{1}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right] + \sum_{i=1}^{I}(1-w_i)\log\left[\frac{\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}{1+\exp[-\boldsymbol{\phi}^T\mathbf{x}_i]}\right]$$

Gradient Based Optimization

- Choose a search direction **s** based on the local properties of the function

- Perform an intensive search along the chosen direction. This is called *line search*

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \left[f[\boldsymbol{\theta}^{[t]} + \lambda \mathbf{s}] \right]$$

- Then set

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} + \hat{\lambda}\mathbf{s}$$

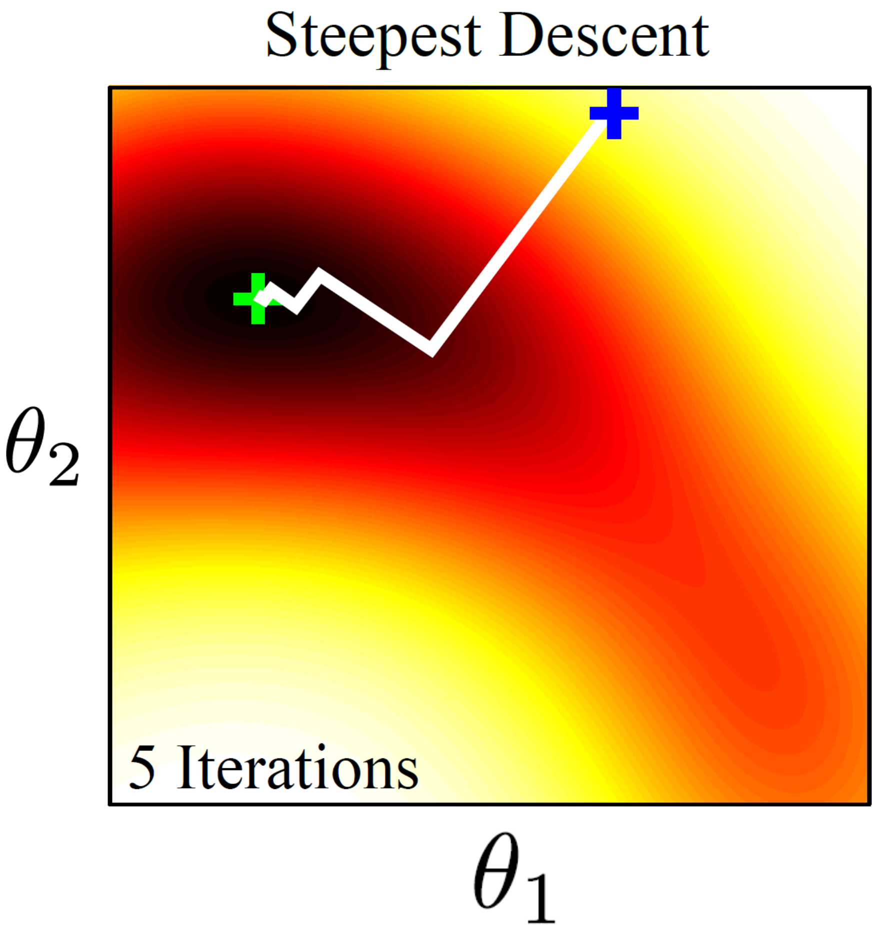
Gradient Descent

Consider standing on a hillside

Look at gradient where you are standing

Find the steepest direction downhill

Walk in that direction for some distance (line search)



Finite differences

What if we can't compute the gradient?

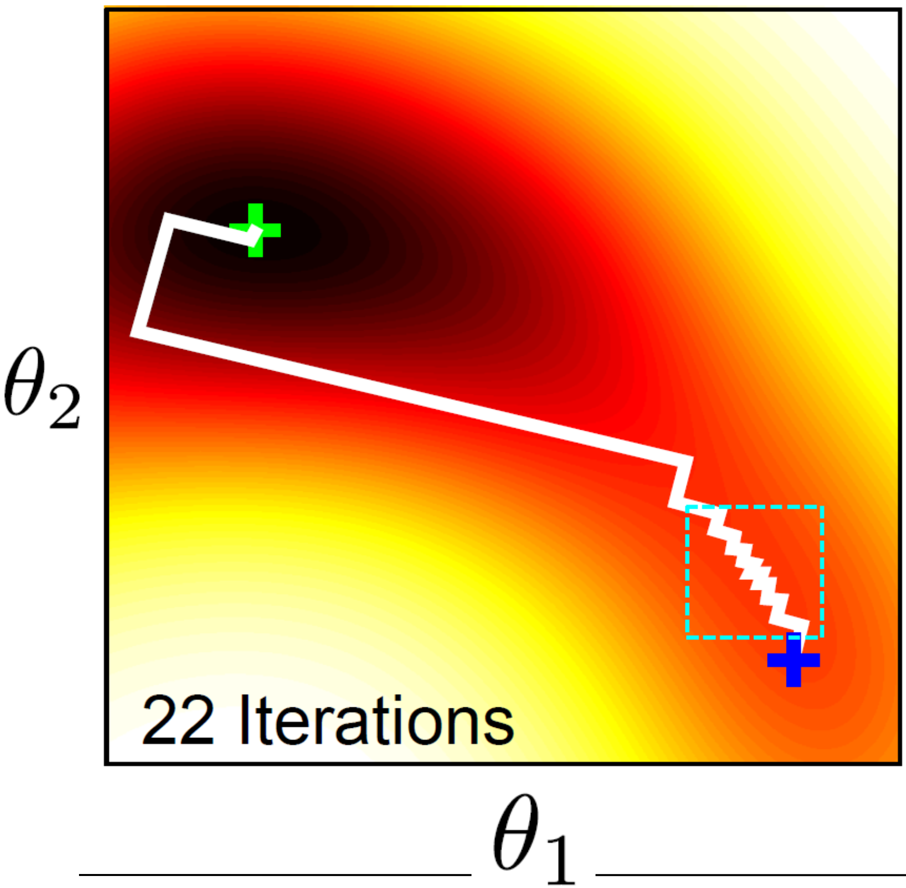
Compute finite difference approximation:

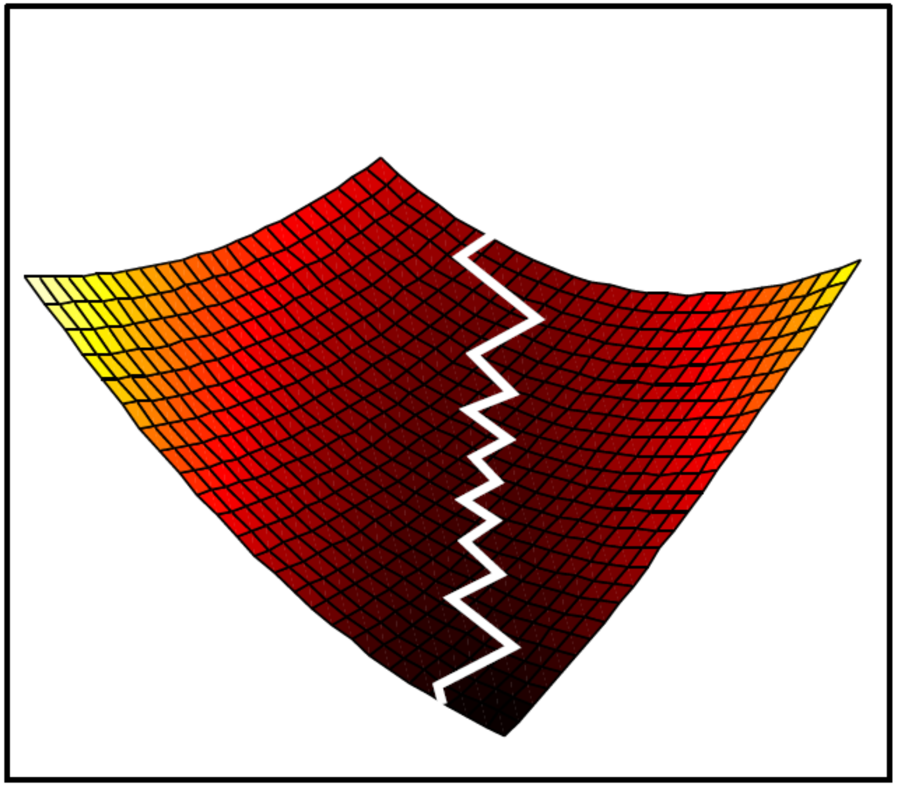
$$\frac{\partial f}{\partial \theta_j} \approx \frac{f[\boldsymbol{\theta} + a\mathbf{e}_j] - f[\boldsymbol{\theta}]}{a}$$

where $\mathbf{e}_j$ is the unit vector in the j^{th} direction

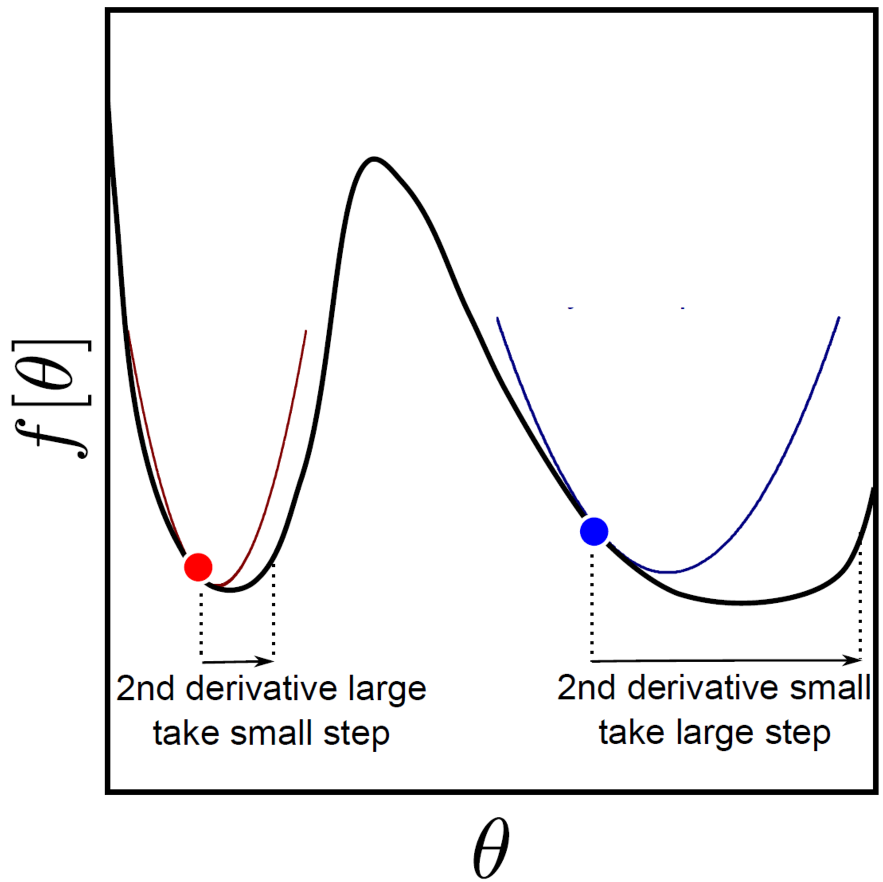
Steepest Descent Problems

Close up





Second Derivatives



In higher dimensions, 2nd derivatives change how much we should move in the different directions: changes best direction to move in.

Newton's Method

Approximate function with Taylor expansion

$$f[\boldsymbol{\theta}] \approx f[\boldsymbol{\theta}^{[t]}] + (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]})^T \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\theta^{[t]}} + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]})^T \left.\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right|_{\theta^{[t]}} (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]})$$

Take derivative

$$\frac{\partial f}{\partial \boldsymbol{\theta}} \approx \left.\frac{\partial f}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{[t]}} + \left.\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}^{[t]}} (\boldsymbol{\theta} - \boldsymbol{\theta}^{[t]}) = 0$$

Re-arrange

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{[t]} - \left(\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right)^{-1} \frac{\partial f}{\partial \boldsymbol{\theta}}$$

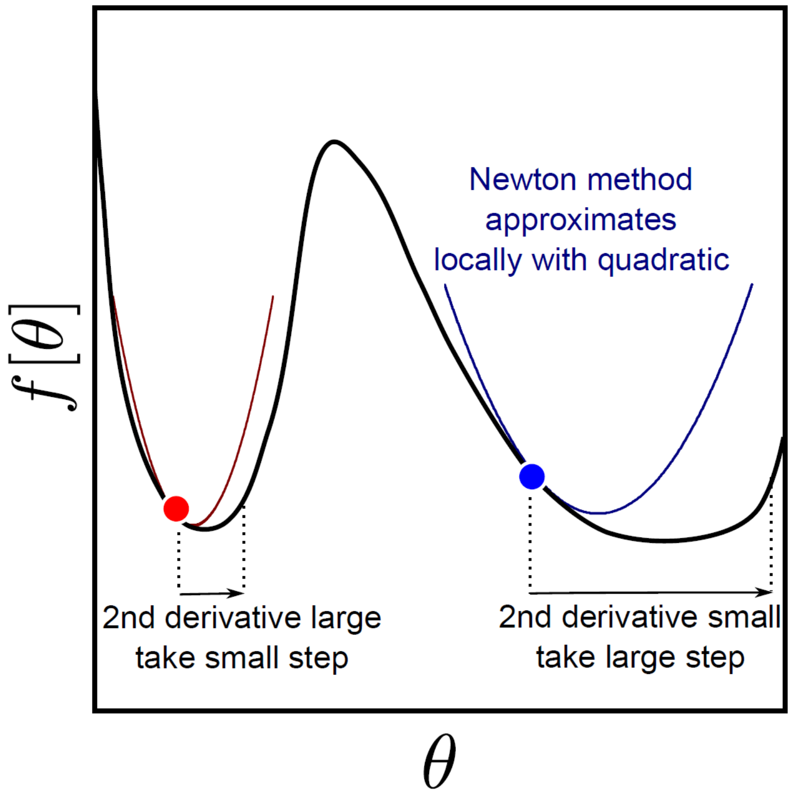
(derivatives taken at time t)

Adding line search

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \lambda \left(\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2}\right)^{-1} \frac{\partial f}{\partial \boldsymbol{\theta}}$$

Newton's Method

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \lambda \left(\frac{\partial^2 f}{\partial \boldsymbol{\theta}^2} \right)^{-1} \frac{\partial f}{\partial \boldsymbol{\theta}}$$

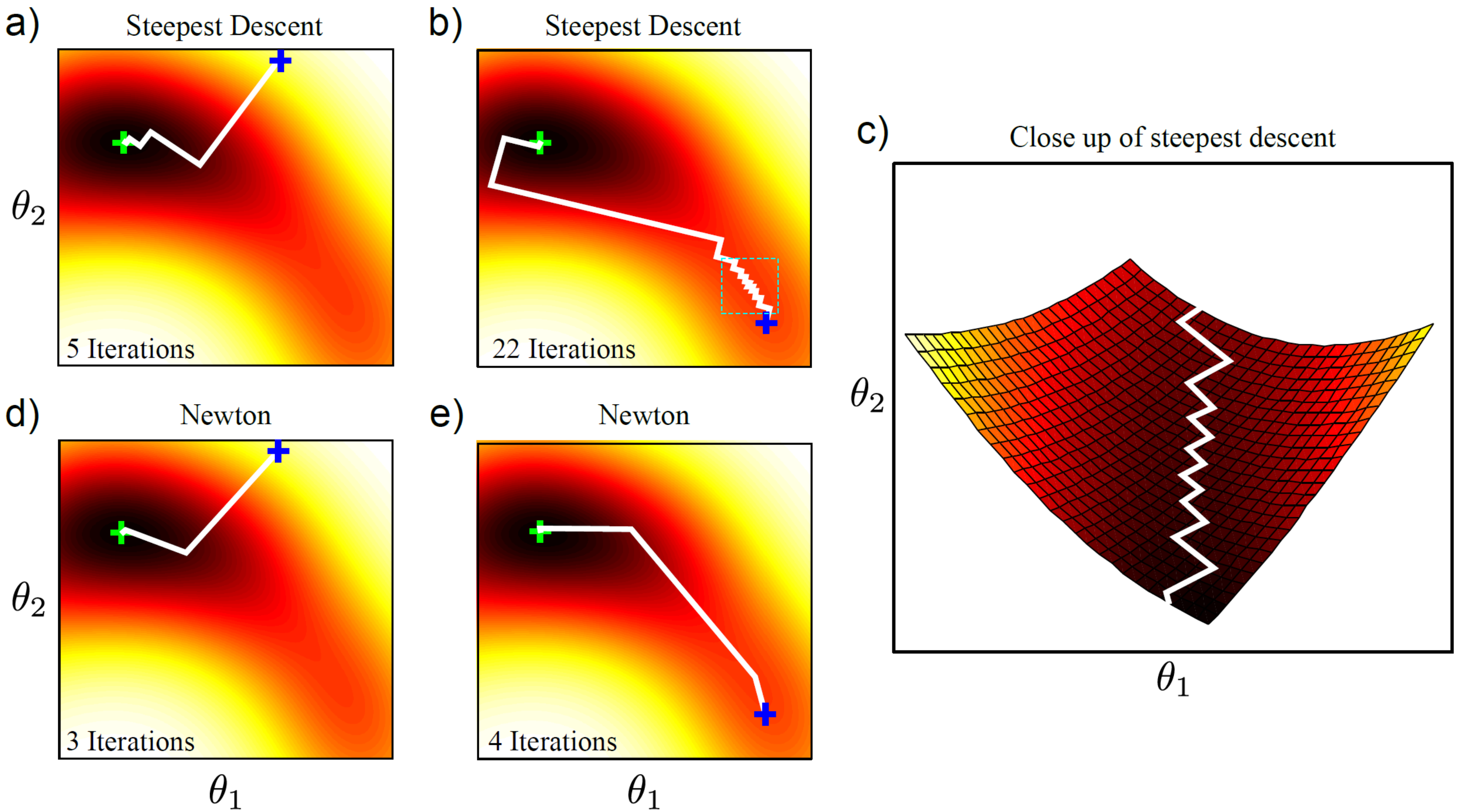


Matrix of second derivatives is called the Hessian.

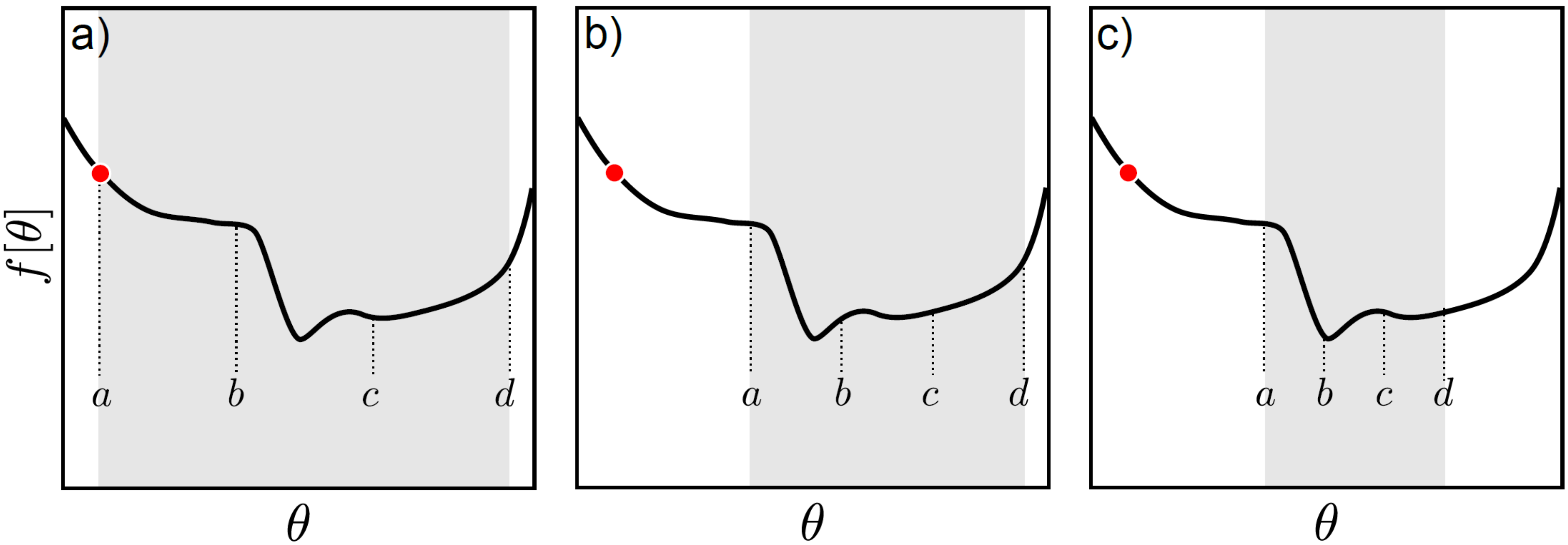
Expensive to compute via finite differences.

If positive definite, then convex

Newton vs. Steepest Descent



Line Search



Gradually narrow down range

Optimization for Logistic Regression

$$\phi^{[t]} = \phi^{[t-1]} + \alpha \left(\frac{\partial^2 L}{\partial \phi^2} \right)^{-1} \frac{\partial L}{\partial \phi}$$

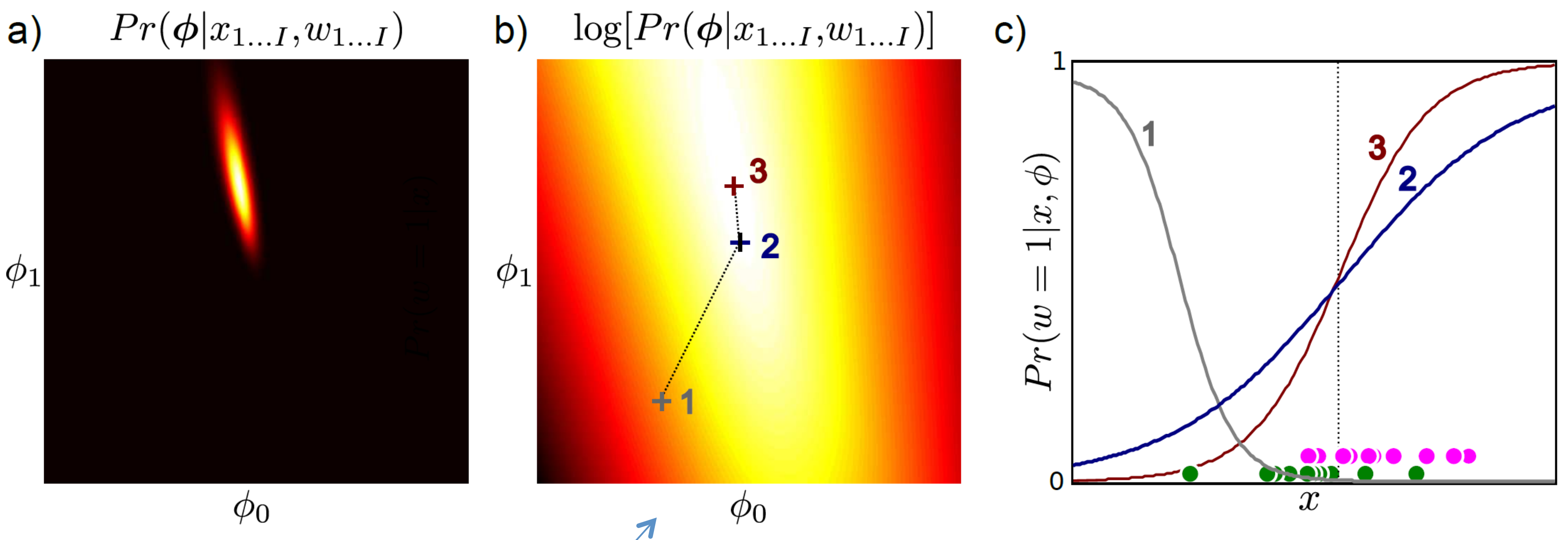
Derivatives of log likelihood:

$$\frac{\partial L}{\partial \phi} = -\sum_{i=1}^{I} (\text{sig}[a_i] - w_i)\mathbf{x}_i$$

$$\frac{\partial^2 L}{\partial \phi^2} = -\sum_{i=1}^{I} \text{sig}[a_i](1 - \text{sig}[a_i])\mathbf{x}_i\mathbf{x}_i^T$$

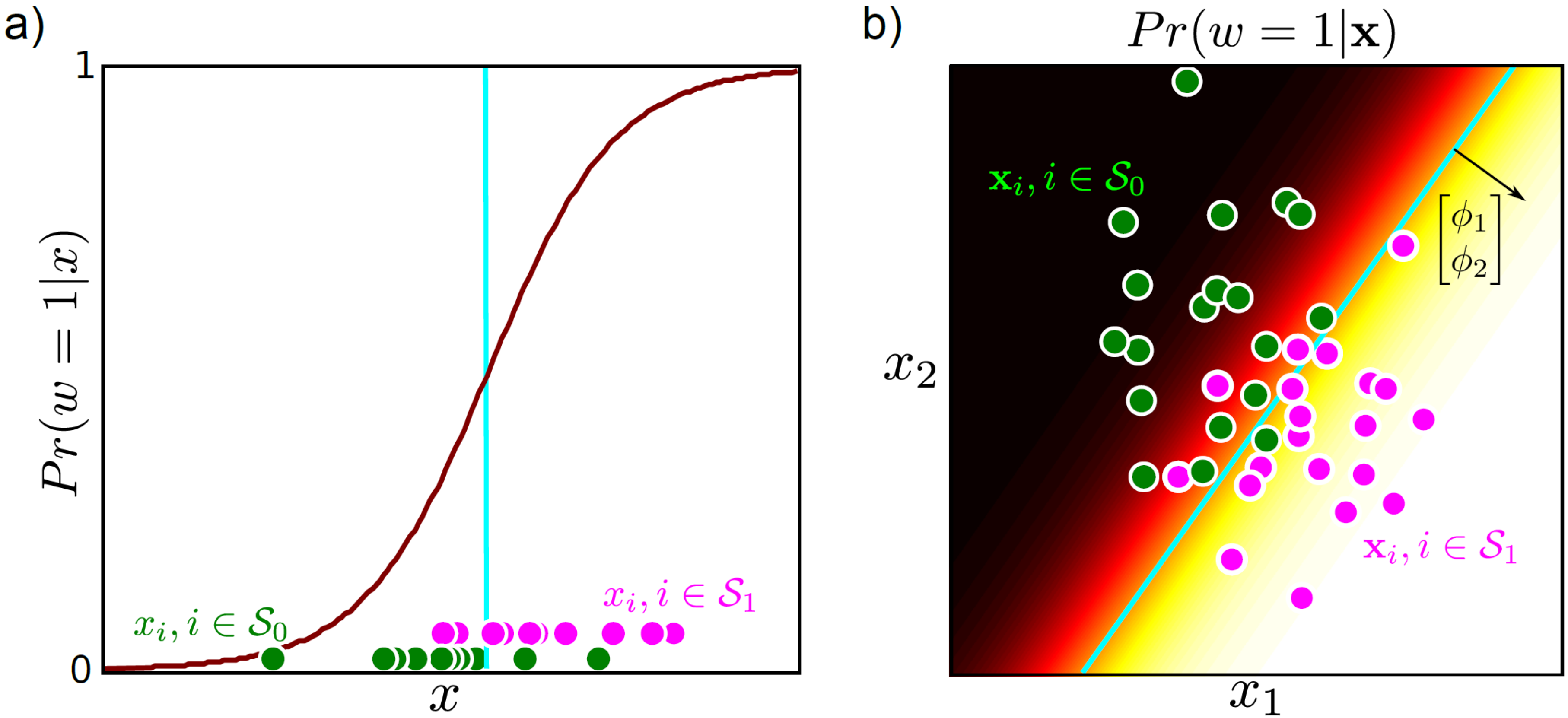
Positive definite!

$$Pr(\mathbf{w}|\mathbf{X},\boldsymbol\phi) = \prod_{i=1}^{I}\left(\frac{1}{1+\exp[-\boldsymbol\phi^T\mathbf{x}_i]}\right)^{w_i}\left(\frac{\exp[-\boldsymbol\phi^T\mathbf{x}_i]}{1+\exp[-\boldsymbol\phi^T\mathbf{x}_i]}\right)^{1-w_i}$$



$$L = \sum_{i=1}^{I} w_i \log\left[\frac{1}{1+\exp[-\boldsymbol\phi^T\mathbf{x}_i]}\right] + \sum_{i=1}^{I}(1-w_i)\log\left[\frac{\exp[-\boldsymbol\phi^T\mathbf{x}_i]}{1+\exp[-\boldsymbol\phi^T\mathbf{x}_i]}\right]$$

Maximum likelihood fits



$$Pr(w|\boldsymbol{\phi}, \mathbf{x}) = \text{Bern}_w \left[\frac{1}{1 + \exp[-\boldsymbol{\phi}^T \mathbf{x}]} \right]$$