

Statistical Machine Learning Overview

Lecture 2

Doug Aberdeen, October 13, 2003

National ICT Australia

Optimisation Methods

Recall

$$\Pr[rain] = w_1 * [clouds = 1, fine = 0] + w_2 * [spring = 1, other = 0].$$

Training is a search for “good” parameter values w_1, w_2 .
Some options:

- Evolutionary Algorithms
- Gradient Methods
- Expectation-Maximisation
- Linear/Quadratic/Semi-definite programming
- Lagrange multipliers

Evolutionary Algorithms

A random search, inspired by evolution in nature.

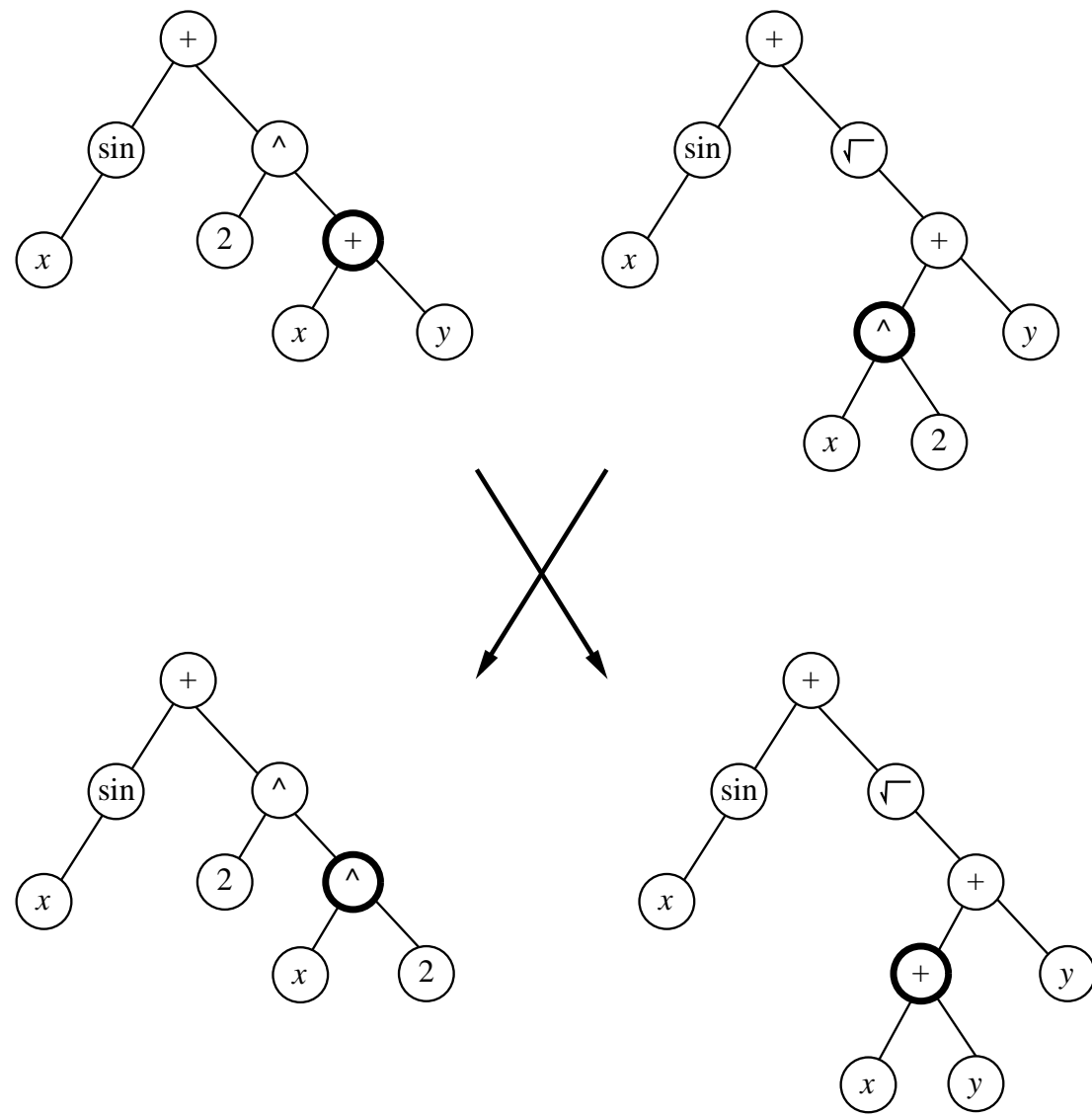
We need:

- A problem representation: trees, programs, bit vectors, strings, grammars, graphs...
- Fitness function, “survival of the fittest”, e.g.
$$E(h_{\mathbf{w}}, \mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{\mathbf{x}} (h_{\mathbf{w}}(x) - y)^2$$
- Mutation and cross-over operators

High-Level EA Alg

1. Initialise population, say 100,000 members
2. Evaluate fitness of members
3. Keep top $x\%$ of members, say 30%
4. Increase population by $y\%$ with mutation, say 5%
5. Fill out population with random crossover
6. While improvement, goto 2.

Example: Genetic Programming



Pro's and Con's of EAs

Pro's

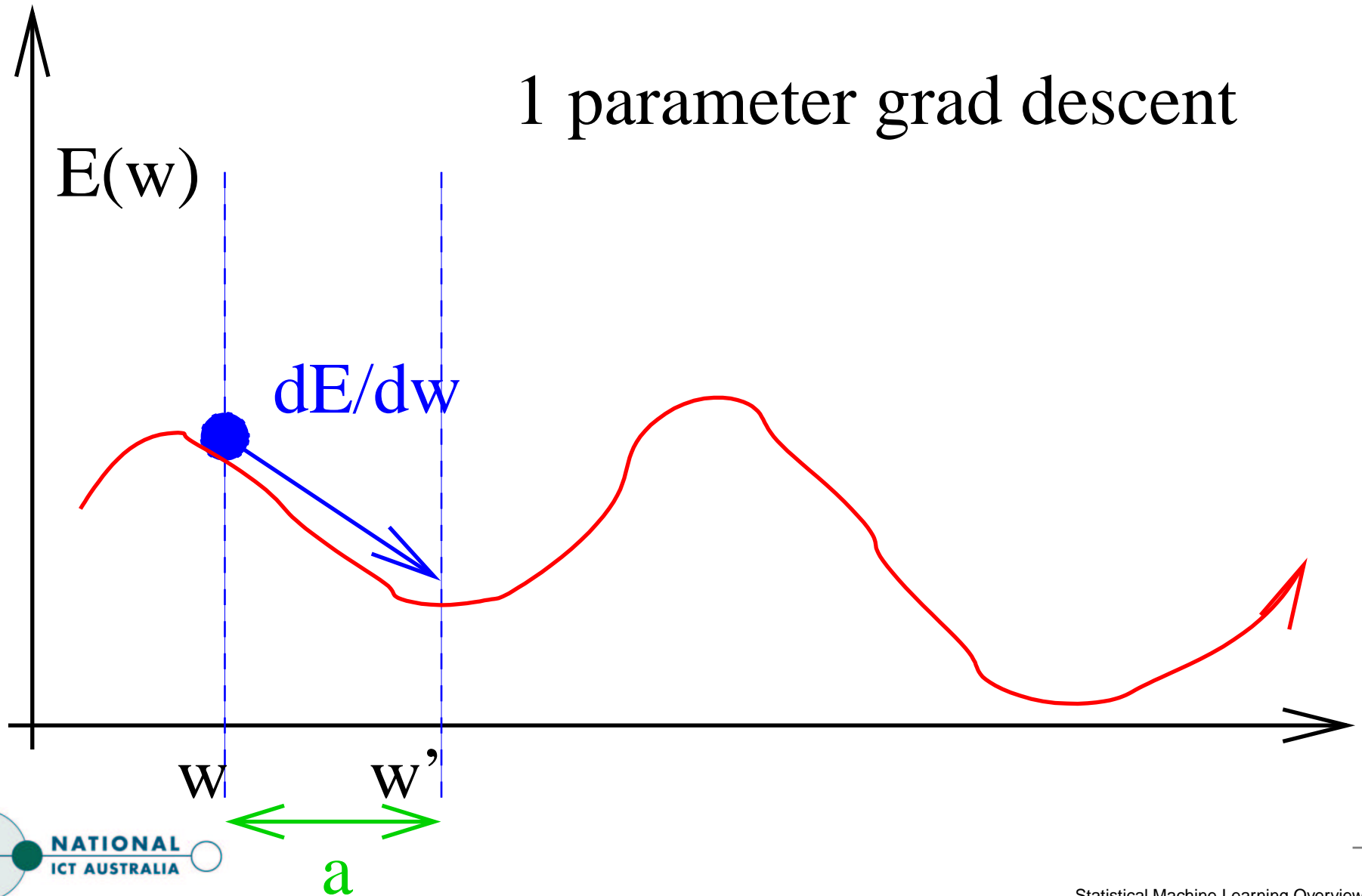
- Successes on very rich data structures, like programs
- Lots of tricks can be learnt from nature, e.g., subspecies
- Mutations reduce chance of getting caught in *local* maxima

Con's

- Can usually do better than random search
- *Conditional* local convergence, e.g., mutation distribution
- Hard to maintain “diversity”

Gradient Descent

1 parameter grad descent



Gradient Descent Alg

Train $h_{\mathbf{w}}(x)$ with error $E(h_{\mathbf{w}}, \mathbf{x}, y)$, training samples \mathbf{x} , step size α , and P parameters \mathbf{w} :

1. Compute or estimate

$$\nabla E(\mathbf{x}, y, \mathbf{w}) = \left[\frac{\partial E(\mathbf{x}, y, \mathbf{w})}{\partial w_1}, \dots, \frac{\partial E(\mathbf{x}, y, \mathbf{w})}{\partial w_P} \right]$$

2. Update parameters $\mathbf{w}' = \mathbf{w} - \alpha \nabla E(\mathbf{x}, y, \mathbf{w})$

3. While $\|\nabla E(\mathbf{x}, y, \mathbf{w})\| > 0$, goto 1

Choosing a Step Size

- Gradient tells us direction to move parameters, not how far
- Fixed step α ?
- What if $\nabla E(\mathbf{x}, \mathbf{y}, \mathbf{w})$ is noisy? E.g., online learning
Perform *stochastic* gradient descent
Ensure $\sum_t \alpha_t = \infty$, and $\sum_t \alpha_t^2 < \infty$
e.g., $\alpha_t = 1/t$
- Line search: double α until $E(\mathbf{x}, \mathbf{y}, \mathbf{w})$ increases

Concave Tricks

- If the error surface is concave, global minimisation is possible
- 2nd derivative (Hessian) can improve the algorithm (Newton's method)
- In practice, approximate local region by 2nd order Taylor expansion
- If the surface is quadratic, optimisation takes one step

Avoiding Local Maxima

“the error surface often looks like a taco shell”
– Gunnar Rätsch

- Momentum terms:

$$\mathbf{w}' = \mathbf{w} - \alpha \nabla_t E(\mathbf{x}, \mathbf{y}, \mathbf{w}) - m\alpha \nabla_{t-1} E(\mathbf{x}, \mathbf{y}, \mathbf{w})$$

- Conjugate gradient: ensure new step direction Δ is orthogonal to all previous directions:

$$\psi = \frac{(\nabla_t E(\mathbf{x}, \mathbf{y}, \mathbf{w}) - \Delta_{t-1}) \cdot \nabla_t E(\mathbf{x}, \mathbf{y}, \mathbf{w})}{\|\Delta_{t-1}\|^2}$$

$$\Delta_t = \nabla_t E(\mathbf{x}, \mathbf{y}, \mathbf{w}) + \psi \Delta_{t-1}$$

When Δ_t is too small, reset $\Delta_t = \nabla_t E(\mathbf{x}, \mathbf{y}, \mathbf{w})$

Repeat training with different initial \mathbf{w}

Estimation-Maximisation (EM)

- Useful for creating ML models of data assuming underlying distribution type
E.g., $\Pr[X, Y | y, \mathbf{w}]$, where
 X = observed data
 Y = unobserved class
 y = assumed class
 \mathbf{w} = parameters describing classes y
- Often used to train maximum likelihood hypothesis:
$$h_{\mathbf{w}}(x) = \arg \max_y \Pr[X = x | y, \mathbf{w}]$$
- Definition: Y is the *hidden* RV; we want to know how likely $X = x$ is assuming $Y = y$ and given parameters \mathbf{w} .
- Still local convergence, but often faster than gradient methods

EM II

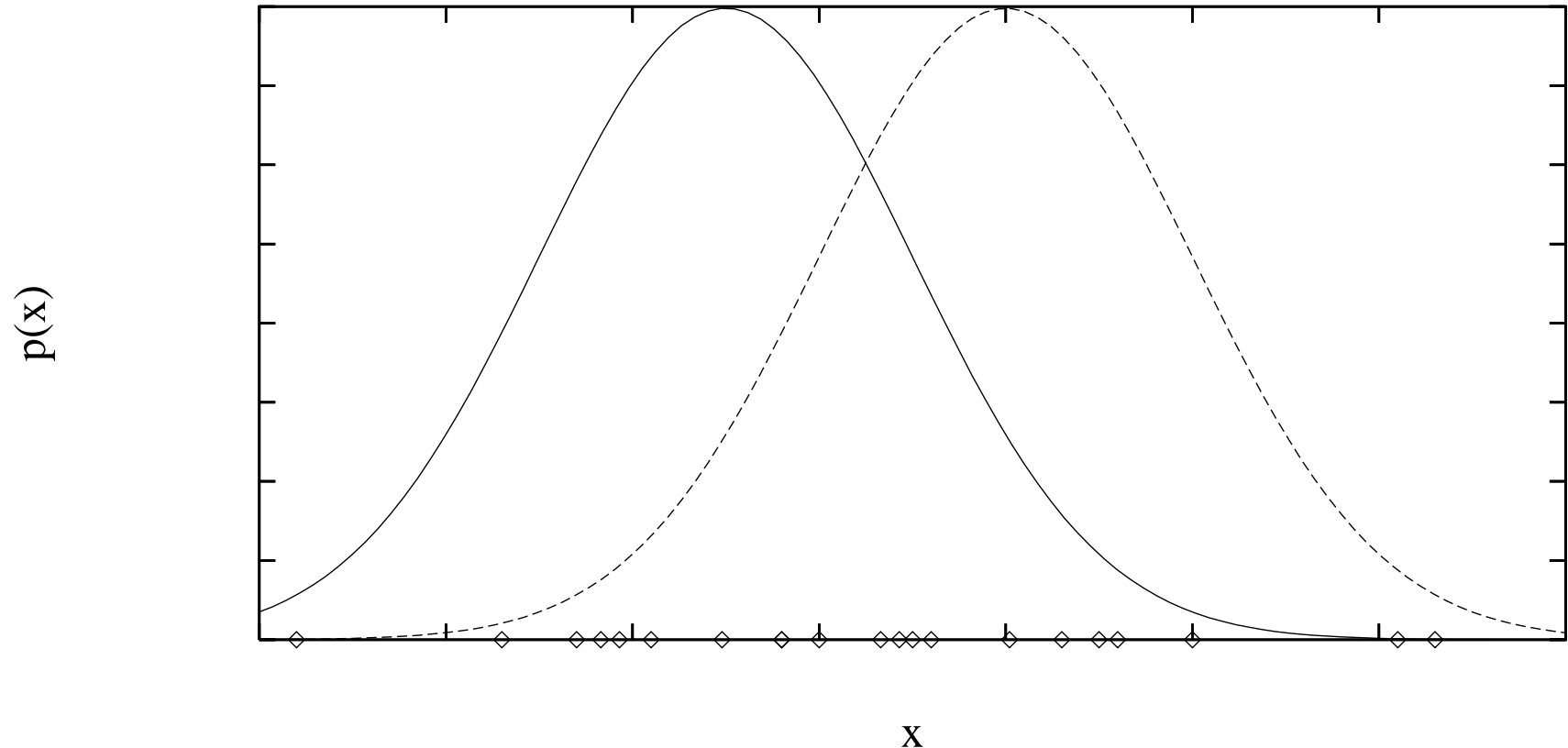
- Let $D = X \cup Y$, and $h = y, w$
- ML hypothesis is the h that maximises $\Pr[D|h]$
- Step 1: Estimate Y assuming h is correct:

$$\mathbb{E}[\ln P(D|h')|h, X]$$

- Step 2: Maximise over possible h' to replace h :

$$h = \arg \max_{h'} \mathbb{E}[\ln P(D|h')|h, X]$$

Example: k-means



X = observed points, Y = unobserved sources, y = true source of points, w = means of sources

Example continued

- Estimation step:

$$\mathbb{E}[d_{ij}] = \frac{\Pr[X = x_i | y = w_j]}{\sum_{n=1}^2 \Pr[X = x_i | y = w_n]} = \frac{e^{0.5\sigma^2(x_i - w_j)^2}}{\sum_{n=1}^2 e^{0.5\sigma^2(x_i - w_n)^2}}$$

- Maximisation step:

$$w_n = \frac{1}{M} \sum_{m=1}^M \mathbb{E}[d_{ij}] x_i$$

More EM Examples

- Determining true state from observations
- Training hidden Markov models: speech recognition
- Building maps from multiple, noisy, data sources
- Modelling spectra for astrophysics