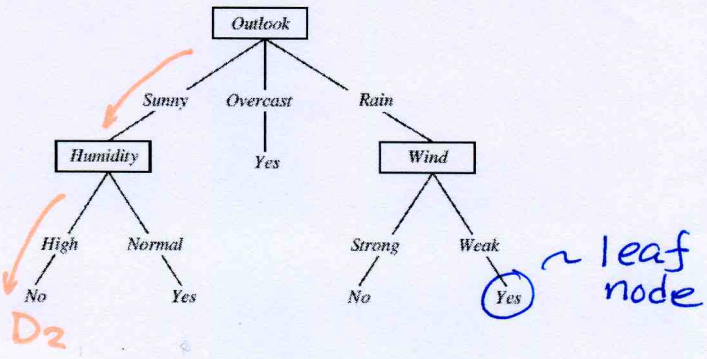
Attribute

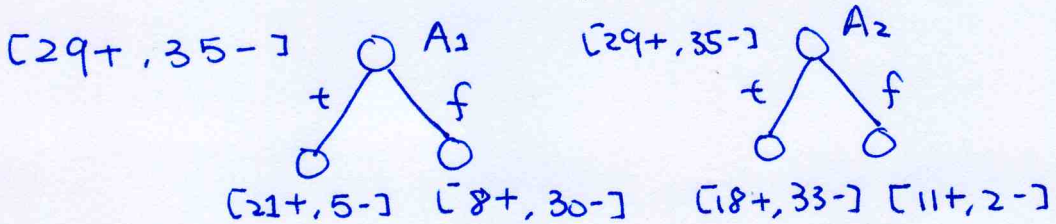
target Attribute [9+, 5-]

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

root node



1. How to decide the root node? Which attribute is better?



how to measure the purity?

- Entropy: $H(S) = -(P_{\oplus} \log P_{\oplus} + P_{\ominus} \log P_{\ominus})$ for $\oplus, \ominus$

  Generally: $H = -\sum_i - P_i \log P_i$

- Information Gain

$$Gain(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

2. Consider the above example:

highted → Humidity

S: [9+, 5-]
E = 0.940

$$H(S) = -\left(\frac{9}{14}\log\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right) = 0.940$$

High → [3+, 4-], E = 0.985

Normal → [6+, 1-], E = 0.592

$$Gain(S, A = Humidity) = 0.940 - \left(\frac{7}{14} \times 0.985\right) - \left(\frac{7}{14} \times 0.592\right) = 0.151$$

Similarly:

Gain(S, Wind) = 0.048

Gain(S, Humidity) = 0.151

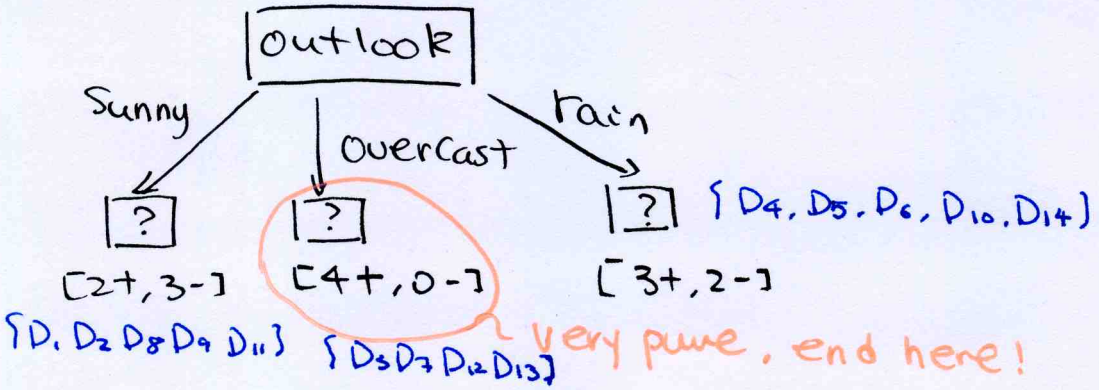
Gain(S, Outlook) = 0.246

Gain(S, temperature) = 0.029

maximum Information Gain

So that: the root node for the "playing-tennis" problem is the attribute of "Outlook"!

3. Decision Tree Building:



$$H(\text{outlook} = \text{Sunny}) = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) = 0.97$$

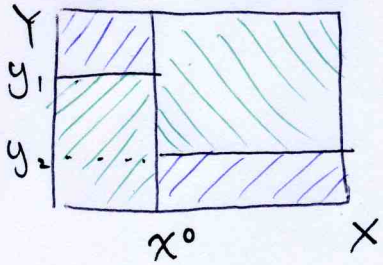
Outlook = Sunny [2+, 3-]

?

{Humidity, Temperature, Wind}

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = 0.97 - \left(\frac{3}{5}\right)\times 0 - \left(\frac{2}{5}\right)\times 0 = 0.97$$

4.



~ split of the space.

IF $x < x^\circ \wedge Y > y_1$ THEN

IF $x < x^\circ \wedge Y < y_1$ THEN

IF $x \geq x^\circ \wedge Y < y_2$ THEN

IF $x \geq x^\circ \wedge Y \geq y_2$ THEN

Decision tree is a transparent model comparing to black-box probabilistic models. A tree can be interpreted into a set of decision rules!

Naive Bayes. $P(T|D) = \frac{P(D|T)P(T)}{P(D)} \propto P(D|T)P(T) = \prod_{d=1}^{4} P(D_d|T)P(T)$

E.g.: $P(T = \text{Yes} | \{\text{Sunny, hot, normal, weak}\}) = ?$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis	
D1	Sunny	Hot	High	Weak	No	*
D2	Sunny	Hot	High	Strong	No	*
D3	Overcast	Hot	High	Weak	Yes	
D4	Rain	Mild	High	Weak	Yes	
D5	Rain	Cool	Normal	Weak	Yes	
D6	Rain	Cool	Normal	Strong	No	
D7	Overcast	Cool	Normal	Strong	Yes	
D8	Sunny	Mild	High	Weak	No	*
D9	Sunny	Cool	Normal	Weak	Yes	△
D10	Rain	Mild	Normal	Weak	Yes	
D11	Sunny	Mild	Normal	Strong	Yes	△
D12	Overcast	Mild	High	Strong	Yes	
D13	Overcast	Hot	Normal	Weak	Yes	
D14	Rain	Mild	High	Strong	No	

$P(\text{sunny}|\text{yes})\,P(\text{hot}|\text{yes})\,P(\text{normal}|\text{yes})\,P(\text{weak}|\text{yes}) = \frac{2}{9}\cdot\frac{2}{9}\cdot\frac{6}{9}\cdot\frac{6}{9}$

$P(\text{sunny}|\text{no})\,P(\text{hot}|\text{no})\,P(\text{normal}|\text{no})\,P(\text{weak}|\text{no}) = \frac{3}{5}\cdot\frac{2}{5}\cdot\frac{1}{5}\cdot\frac{2}{5}$

$P(\text{yes}) = \frac{9}{14}$

$P(\text{no}) = \frac{5}{14}$.

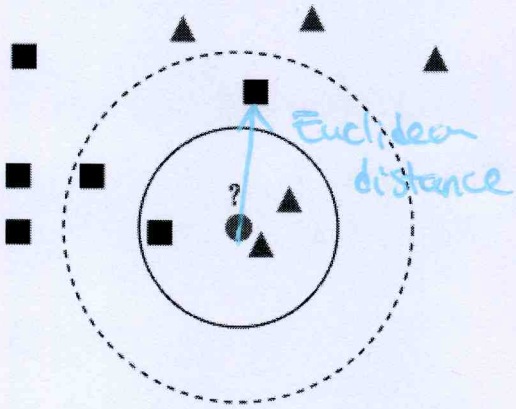
$P(T=\text{yes}|X) = \frac{2}{9}\cdot\frac{2}{9}\cdot\frac{6}{9}\cdot\frac{6}{9}\cdot\frac{9}{14} = \boxed{0.0141}$

$P(T=\text{No}|X) = \frac{3}{5}\cdot\frac{2}{5}\cdot\frac{1}{5}\cdot\frac{2}{5}\cdot\frac{5}{14} = 0.0068$

$P(\text{yes}|x) > P(\text{No}|x)$

① K-Nearest Neighbour (instance-based learning, lazy learning)

KNN for classification, see the following graph.

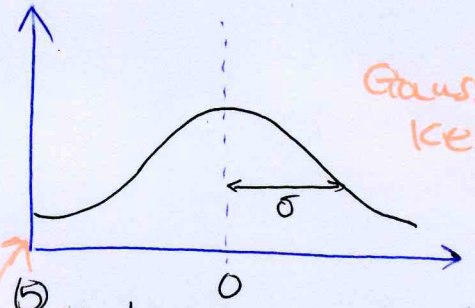


② Radial Basis Function

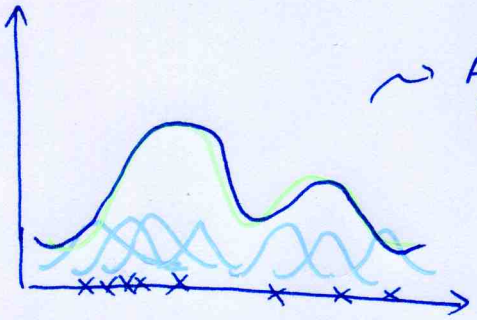
$$\hat{f}(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x))$$

$K_u(d(x_u, x))$ is a kernel function.

$$K_u(d(x_u, x)) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} d(x_u, x)^2\right)$$



③ kernel density estimation.



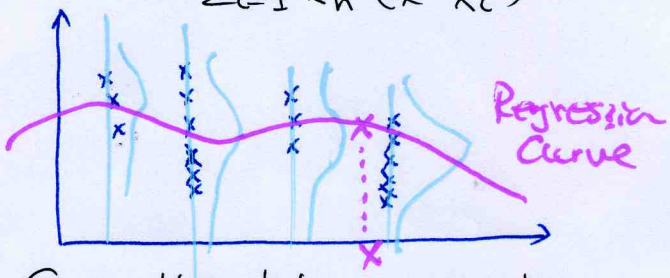
~ Addiative kernels.

It gives more smooth estimation of the sampling data elements!

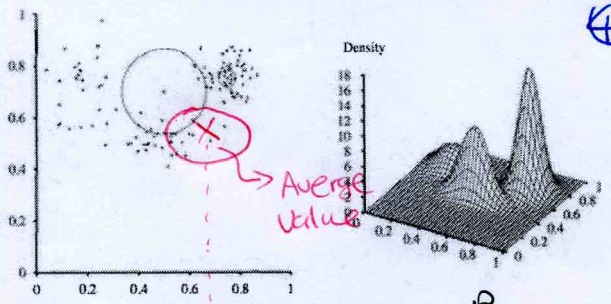
$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i)$$

⑤ Nadaraya-Watson kernel Regression: to estimate Conditional Expectation $E[Y|X]$

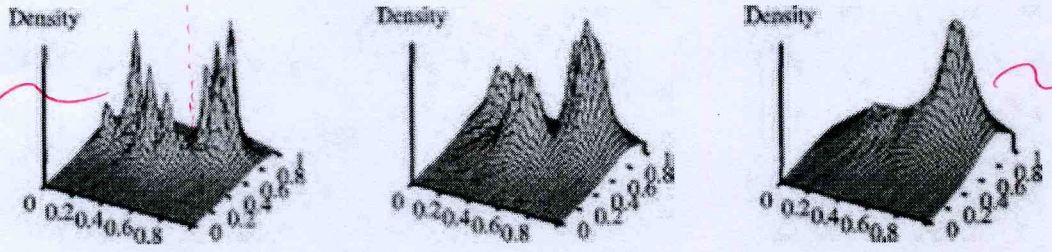
$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)}$$



④ Given the data elements sampled from Fig. B. are shown in Fig. A. Fig C.D and E are the K-NN density estimation with K=3, 10 and 40.



A B



C D E

KNN Regression, ~~data~~ data element x: $x = \frac{1}{k} \sum_{i=1}^{k} N(x_i, x)$

where $N(x_i, x)$ is the nearest k data elements given x.

① $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ Cost Function

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ ← gradient descent

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$
$$= 2 \cdot \frac{1}{2}(h_\theta(x^{(i)} - y^{(i)}) \frac{\partial}{\partial \theta_j}(h_\theta(x^{(i)}) - y^{(i)})$$
$$= (h_\theta(x^{(i)}) - y^{(i)}) \frac{\partial}{\partial \theta_j}(\sum_{i=0}^{n} \theta_i x_i^{(i)} - y^{(i)})$$
$$= (h_\theta(x^{(i)}) - y) x_j^{(i)}$$

$$\Rightarrow \theta_j = \theta_j - \alpha (h_\theta(x^{(i)}) - y) x_j^{(i)}$$
$$= \theta_j + \alpha (y - h_\theta(x^{(i)}) x_j^{(i)}$$

~ LMS Rule
Widrow-Hoff learning Rule

$$\begin{cases} h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ h_\theta(x) = \sum_i \theta_i x_i = \theta^T x \end{cases}$$

② Batch gradient descent

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

for every $j = 1 \cdots n$ (dimension of x)

② Incremental gradient descent

for $i = 1$ to m: { $\theta_j := \theta_j + \alpha (y - h_\theta(x^{(i)})) x_j^{(i)}$ }

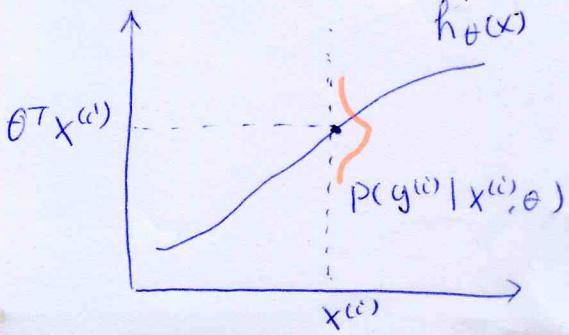
③ Probabilistic Interpretation.

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$
$$\varepsilon^{(i)} = (y^{(i)} - \theta^T x^{(i)})$$

If we assume: $p(\varepsilon^{(i)}) \sim N(0, \sigma^2)$

$$p(y^{(i)} | x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$y^{(i)} | x^{(i)}, \theta \sim N(\theta^T x^{(i)}, \sigma)$$



Given the probabilistic interpretation of the LMS. We can regard the linear regression as maximum likelihood estimation.

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)} | x^{(i)}, \theta)$$

$$y^{(i)} | x^{(i)}, \theta \sim N(\theta^T x^{(i)}, \sigma)$$

using log likelihood:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \log P(y^{(i)} | x^{(i)}, \theta)$$

$$= \sum_{i=1}^{m} \log \left\{ \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2\sigma^2} (y^{(i)} - \theta^T x^{(i)})^2 \right\} \right\}$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2$$

maximize log liklihood $\ell(\theta)$ is equivalent to minimize $\sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2$, which is exactly LMS.

④ In vector form (or in matrix)

Error Term: $E(\theta) = (y - \theta^T x)(y - \theta^T x)^T$

$$= (y - \theta^T x)(y^T - x^T \theta)$$

$$= yy^T + \theta^T x x^T \theta - \theta^T x y^T - y x^T \theta$$

(1×N, N×d, d×1)

Some descriptions (dimensions) of variables.

y : 1×N	y^T : N×1
θ : d×1	θ^T : 1×d
x : d×N	x^T : N×d

yy^T: 1×N, N×1

$\theta^T x x^T \theta$: 1×d, d×N, N×d, d×1

$\theta^T x y^T$, $y x^T \theta$: 1×d, d×N, N×1

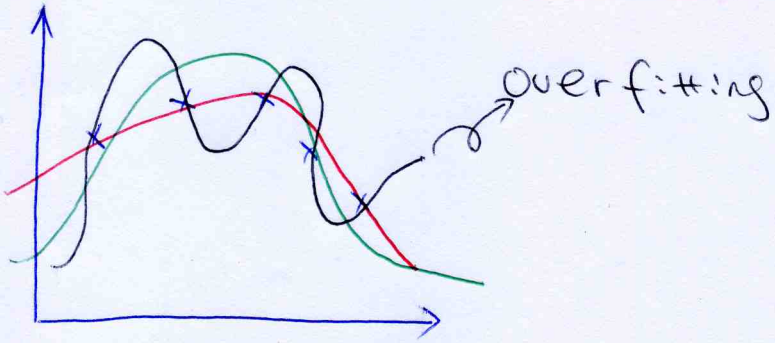
All the terms are scalars.

To take the derivative:

$$\frac{\partial E}{\partial \theta} = 0 \Rightarrow 2 x x^T \theta - 2 x y^T = 0$$

$$\Rightarrow \theta = (x x^T)^{-1} x y^T$$

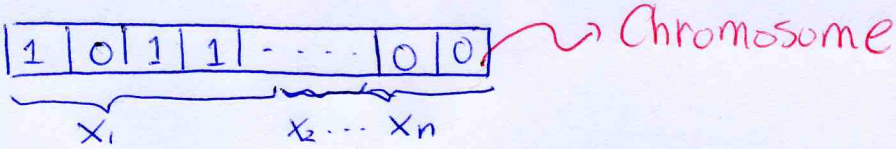
LMS estimation



Genetic Algorithm. ~ Stochastic optimization technique

For a complex function $f(x) = f(x_1, x_2 \cdots x_n)$ and $\{x_1, \cdots x_n\}$ following some constraints $g(x_1 \cdots x_n) = 0$ how to find the maximum value of the given function.

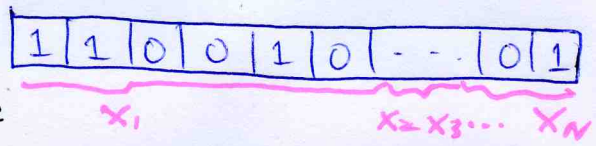
1. Encoding



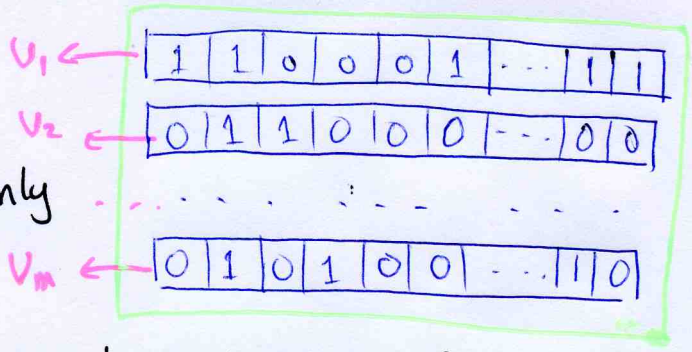
Each variable can be represented by a binary code.

2. Fitness function.

Evaluate a chromosome by $f(x_1 \cdots x_N)$



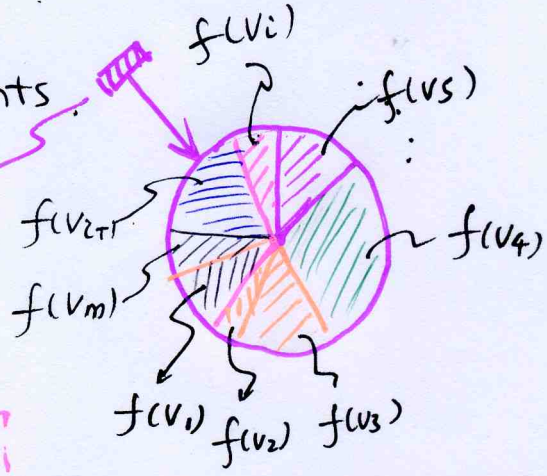
3. A pool of chromosomes. which are generated randomly

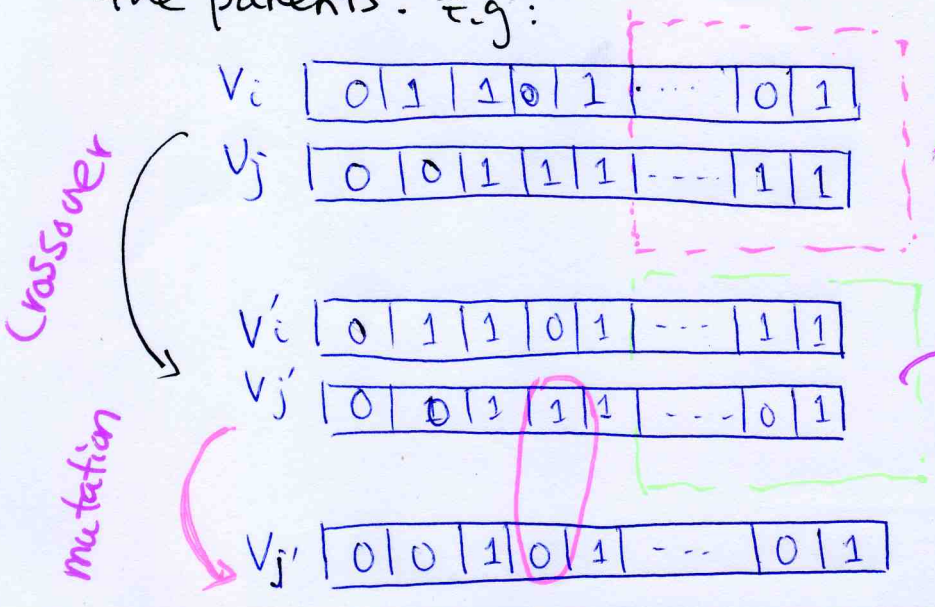


Each chromosome can be evaluated by the fitness function. $\{f(V_1), f(V_2) \cdots f(V_m)\}$

4. Ranking and selecting good parents

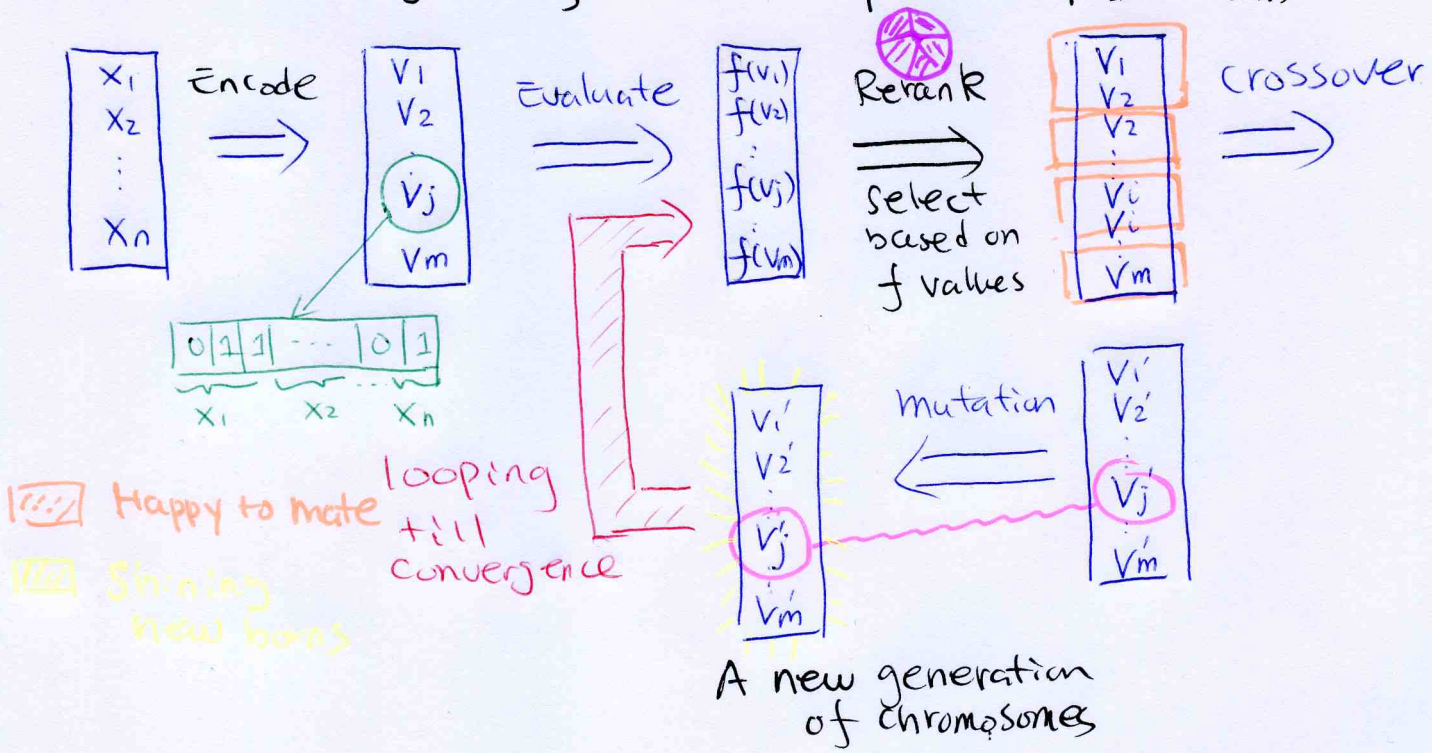
The chromosomes with bigger $f(V_i)$ values have better chances to be selected as the parents. E.g:



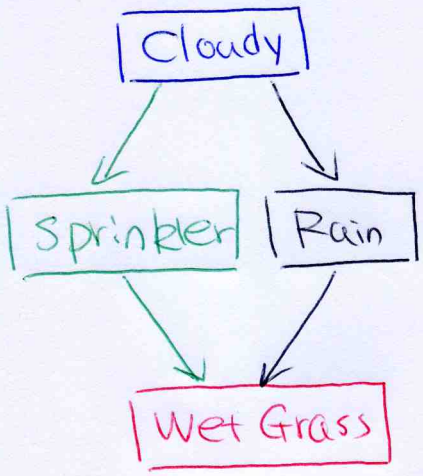


5. GA process.

Initialize a pool of chromosomes $V_1 \cdots V_m$, where V_i is encoded by binary codes to represents $\{X_1 \cdots X_n\}$



Bayesian Network (Variable Elimination method)



prior for Cloudy (C)

T_1 $P(C=F) = P(C=T) = 0.5$

T_2

C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

T_3

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

$P(R=T|C=T) = 0.8$

Reasoning Table for Wet Grass:

T_4

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Suppose all the 3 tables are given from training. How do we do the inference?

E.g. $P(S=T|W=T) = ?$

Using Bayesian Theorem:

① $P(S=T|W=T) = \frac{P(W=T|S=T)P(S=T)}{P(W=T)}$ = ⑤ $\frac{P(W=T, S=T)}{P(W=T)}$ ②

② $P(W=T, S=T) = \sum_c \sum_r P(C=c, S=T, R=r, W=T)$

$= \sum_c P(C=c) P(S=T|C=c) \sum_r P(R=r|C=c) \cdot P(W=T|S=T, R=r)$ ③

③ $\sum_r P(R=r|C=c) \cdot P(W=T|S=T, R=r)$

$= P(R=T|C=c) \cdot P(W=T|S=T, R=T) + P(R=F|C=c) \cdot P(W=T|S=T, R=F)$

$= P(R=T|C=c) \times 0.99 + P(R=F|C=c) \times 0.9$

②~③ ⇒ ④ $P(W=T, S=T) = \sum_c P(C=c) P(S=T|C=c) \times (0.99 P(R=T|C=c) + 0.9 P(R=F|C=c))$

④: $P(W=T, S=T) = P(C=T) P(S=T|C=T)(0.99 P(R=T|C=T) + 0.9 P(R=F|C=T))$

$+ \cancel{0.5} P(C=F) P(S=T|C=F)(0.99 P(R=T|C=F) + 0.9 P(R=F|C=F))$

$= 0.5 \times 0.1 \times (0.99 \times 0.8 + 0.9 \times 0.2) + 0.5 \times 0.5 \times (0.99 \times 0.2 + 0.9 \times 0.8)$

$= 0.05 \times 0.972 + 0.25 \times 0.918 = 0.2781$

⑤ $P(W=T) = \sum_c \sum_r \sum_s P(C=c, S=s, R=r, W=T)$

$= \sum_c P(C=c) \sum_s P(S=s|C=c) \sum_r P(R=r|C=c) P(W=T|S=s, R=r)$

{C} {C, S} {C, S, R}

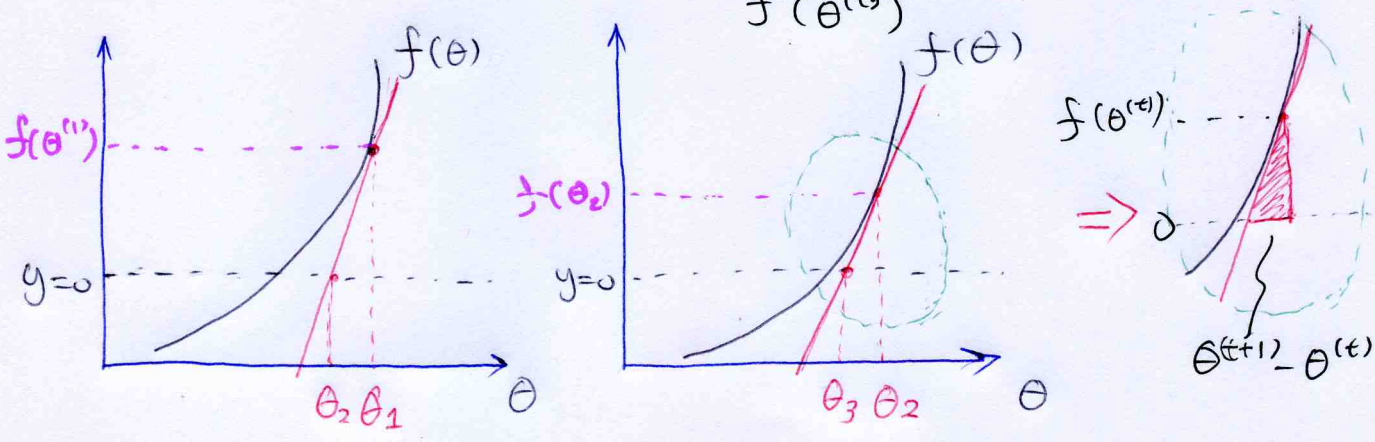
Newton's Method for maximizing likelihood.

① Newton's method for finding a zero of a function.

$f: \mathbb{R} \to \mathbb{R}$, and wish to find the value of θ, so that $f(\theta)=0$, $\theta \in \mathbb{R}$ is a real number. Then θ is updated based on:

$$\theta \leftarrow \theta - \frac{f(\theta)}{f'(\theta)}$$

or formally: $\theta^{(t+1)} \leftarrow \theta^{(t)} - \frac{f(\theta^{(t)})}{f'(\theta^{(t)})}$



$$\frac{\theta^{(t+1)} - \theta^{(t)}}{f(\theta^{(t)})} = \frac{1}{f'(\theta^{(t)})}$$

② In maximizing likelihood.

$\frac{\partial \ell}{\partial \theta}=0$, so that $\theta^{(t+1)} \leftarrow \theta^{(t)} - \frac{\ell'(\theta^{(t)})}{\ell''(\theta^{(t)})}$

To generalize the newton's method.

(Newton - Raphson Method) is given by:

$$\vec{\theta}^{(t+1)} \leftarrow \vec{\theta}^{(t)} - H^{-1} \nabla_\theta \ell(\theta)$$

where H is the Hessian matrix

$$H_{ij} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j} \qquad \nabla_\theta \ell(\theta) = \frac{\partial(\theta)}{\partial \theta_i} \quad \text{for } i = 1 \cdots d$$

③ Newton method is good for convex function maximization.

$f: \mathbb{R}^n \Rightarrow \mathbb{R}$ is convex if

$$f(\theta x + (1-\theta) y) \le \theta f(x) + (1-\theta) f(y) \quad (0 \le \theta \le 1)$$