Phân lớp và hồi quy (Classification and Regression)

Hoàng Xuân Huấn

Nội dung

- Phát biểu bài toán
- Phân lớp nhờ hàm quyết định: Cực tiểu khoảng cách, SVM
- Phân lớp Bayes
- Cây quyết định: ID3, C45, CART
- Tìm hàm hồi quy: Phương pháp gradient

Phát biểu bài toán

- ***Bài toán phân lớp*** (Classification/ taxonomy)
  + Phân lớp: xếp các đối tượng thành nhóm dựa trên tính tương tự của chúng

  + ***Phân loại*** (Categorize): Học có giám sát, dựa trên tập mẫu hoặc thông tin bổ trợ
  Ví dụ: Đánh giá ý kiến, phân loại thông tin, nhận dạng hình ảnh,chữ viết…

  +Phân cụm (cluster): Học không giám sát.

  +Phần tử ngoại lai (Outlier)
- ***Bài toán hồi quy*** (Regression):
  +Xác định giá trị một hàm cho các đối tượng dựa trên một tập đã quan sát được
  + Phân tích hồi quy : Phương pháp giải tich cho các đối tượng có đặc trưng là vectơ thực

Phát biểu bài toán

Bài toán. Cho tập đối tượng **X** , tập nhãn **Y** và **t**ập dữ liệu quan sát được:

D= $\left\{(x^k, y^k)\right\}_{k=1}^{N}; x^k \in X, y^k \in Y\ \forall k$

Cần tìm nhãn c(***x***) cho các $x \in X$

- Bài toán phân lớp: ***Y*** là tập hữu hạn

  Chú ý! Trong phân lớp thống kê, có thể cho các ước lượng xác suất thay cho tập D

- Bài toán hồi quy: ***Y*** là tập số thực, các nhãn *y* của ***x*** là giá trị của hàm *f(**x**)* chưa biết.

  Cần tìm hàm hồi quy *g*(***x***) để đoán nhận *f(**x**)*

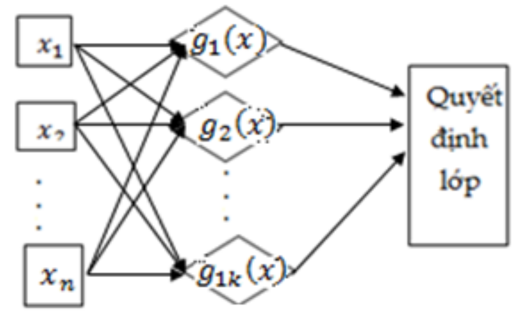
  g(***x***) có thể cho bởi tập luật hoặc biểu thức giải tích

Phân lớp nhờ hàm quyết định

Gỉa sử có k lớp **Y**= $\{\omega_1, \ldots, \omega_k\}$; Với mỗi $i \leq k$:

- Xây dựng hàm g_i tính mức độ $g_i(\mathbf{x})$ thuộc ω_i cho mỗi $x \in X$
- Quyết định $x \in \omega_i \ khi \ g_i(x) = \max\{g_j(x) / j \leq k\}$

  g_i được gọi là ***hàm quyết định*** của lớp *I*



Các hàm phân biệt

- *Ví dụ.* g_i có thể là:

  +Số từ khóa thích hợp với loại băn bản

  + Độ giống của hình ảnh, chữ, vân tay…
- Nếu đối tượng có đặc trưng là vectơ thực n-chiều, các hàm quyết định phân $\boldsymbol{R^n}$ thành các ***miền quyết định*** $\boldsymbol{R_i}$ $R_i = \{x \in X : g_i(x) = \max\{g_j(x) / j \leq k\}\}$
- **P**hần giao nhau gọi là ***biên quyết định***

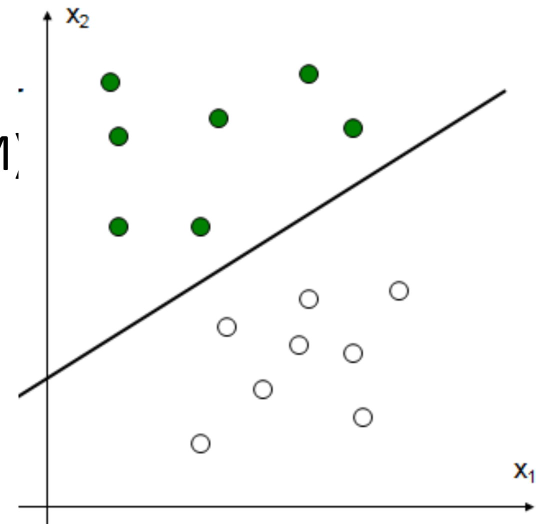
Phân lớp nhờ hàm quyết định

- Khi có 2 lớp: Chỉ cần dùng 1 hàm quyết định $g(\boldsymbol{x})=g_1(\boldsymbol{x})-g_2(\boldsymbol{x})$

  Quy tắc qđ: $x \in \omega_1$ nếu $g(\boldsymbol{x}) > 0$, ngược lại $x \in \omega_2$
- Phân biệt tuyến tính: $g_i(x) = \sum_{i=1}^{n} w_j^i x_i + w_0^i = \left(w^i\right)^T x + w_0^i = \left\langle w^i, x \right\rangle + w_0^i$

+Phân biệt khoảng cách cực tiểu

+ Máy véctơ tựa (Support Vector Machine SVM)



Phân lớp khoảng cách cực tiểu

- Khoảng cách (mêtric) Mahalanobis $d_A(x,y)$ cho bởi ma trận A=$(a_{i.j})_{n\times n}$:

$$d_A^2(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}(x_i - y_i)(x_j - y_j) = (x-y)^T A(x-y) = \langle A(x-y), x-y \rangle$$

Thường dùng A=I (ma trân đơn vị) hoặc $A= C^{-1}$ (C là ma trận hiệp phương sai)

+ Tính tâm $\boldsymbol{m_i}$ của ω_i

+Hàm quyết định $\boldsymbol{g}_i$ sẽ là:

$$\boldsymbol{g}_i(\boldsymbol{x}) = -d_A^2(x, m_i) = -\langle A(x-m_i), x-m_i \rangle = -\langle Ax, x \rangle + 2\langle Am_i, x \rangle - \langle Am_i, m_i \rangle$$

Đều chứa số hạng $-\langle Ax, x \rangle$ nên bỏ đi và thay bởi:

$$\boldsymbol{g}_i(\boldsymbol{x}) \leftarrow \langle Am_i, x \rangle - 0{,}5\langle Am_i, m_i \rangle = m_i^T A m_i - 0{,}5 m_i^T A x$$

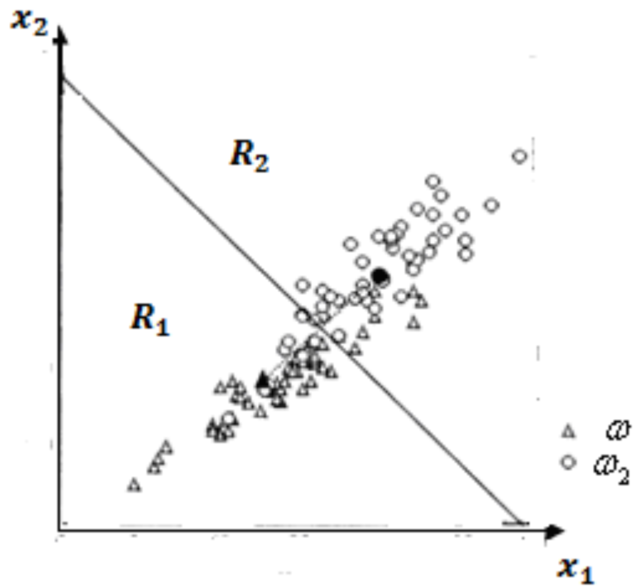
là phân biệt tuyến tính

Chú ý!

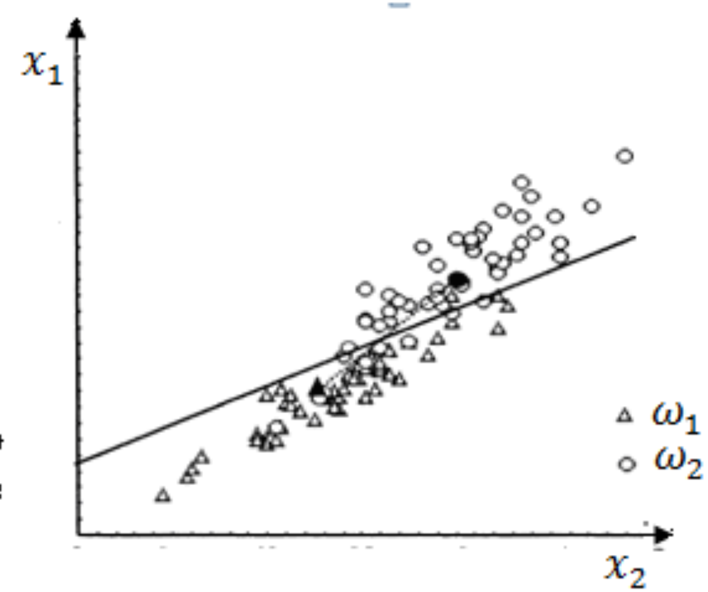
+Khi dùng mêtric Euclide thì biên quyết định của là siêu phẳng trực giao với đoạn nối hai tâm của mỗi lớp

+ Khi dùng $A=C^{-1}$, nếu tính riêng ma trận hiệp phương sai cho từng lớp để tính khoảng cách thì kết quả phân lớp tốt hơn

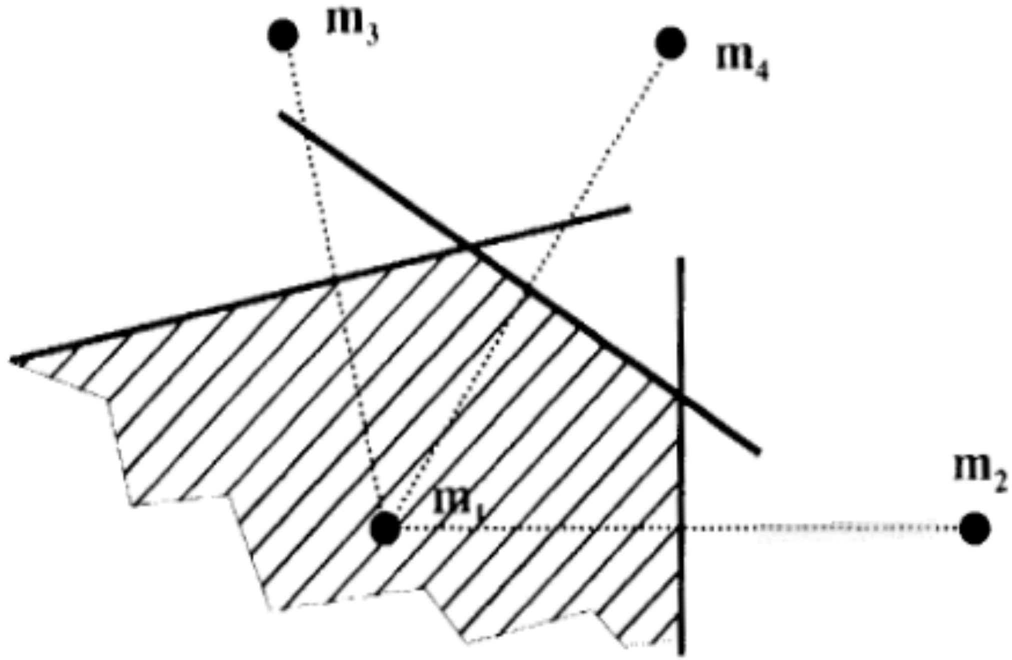
a) A=I (mêtric Euclide)



b) $A = C^{-1}$



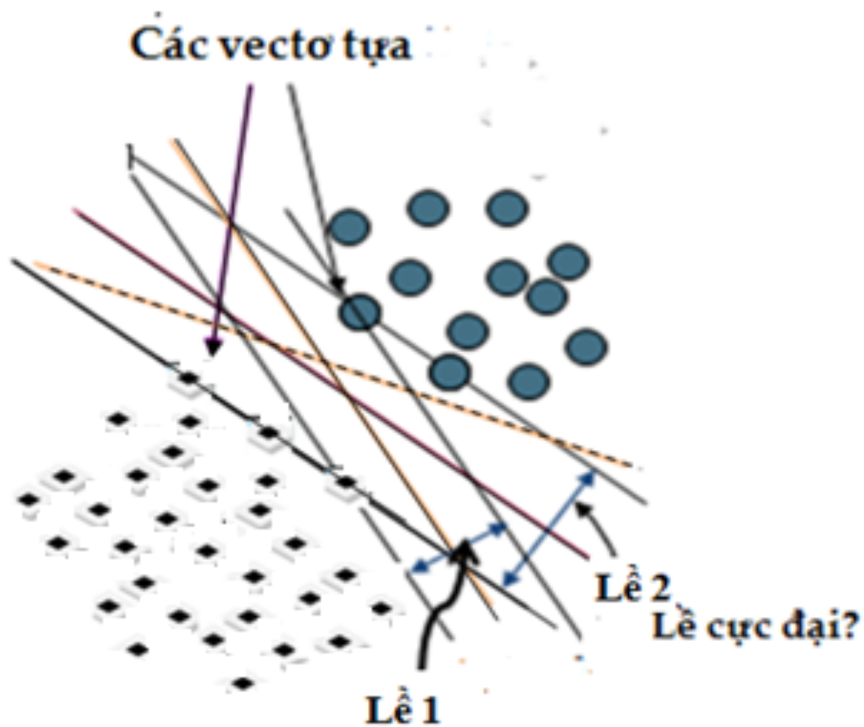
Miền quyết định của có phân biệt tuyến tính với 3 lớp



Máy véctơ tựa

Các khái niệm:

- Tách được tuyến tính
- Siêu phẳng tựa
- Vecto tựa
- Lề



Khi các lớp tách được tuyến tính

- Xét trường hợp có 2 lớp với nhãn tương ứng là -1/ +1.
- Tập mẫu là D = $\{(\boldsymbol{x}^t, y^t)/t=1,..N\}$,
  trong đó $\mathbf{y}^t = +1$ nếu $\mathbf{x}^t \in \omega_1$ và $y^t = -1$ nếu $x^t \in \omega_2$

Ta sẽ tìm $\boldsymbol{w}$ và w_0 sao cho

$$\boldsymbol{g}(\boldsymbol{x}^t) = \boldsymbol{w}^T\boldsymbol{x}^t + w_0 \geq +1 \text{ khi } y^t = +1$$

$$\boldsymbol{g}(\boldsymbol{x}^t) = \boldsymbol{w}^T\boldsymbol{x}^t + w_0 \leq -1 \text{ khi } y^t = -1,$$

hay có thể viết lại là

$$y^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) \geq 1$$

Co giãn w, w_0 để $\boldsymbol{g}(\boldsymbol{x}^t) = \begin{cases} 1 \text{ nếu } \boldsymbol{x} \in \omega_1 \\ -1 \text{ nếu } x \in \omega_2 \end{cases}$ ta có:

Lề của siêu phẳng là $\frac{1}{|w|} + \frac{1}{|w|} = \frac{2}{|w|}$

Dẫn đến bài toán tối ưu

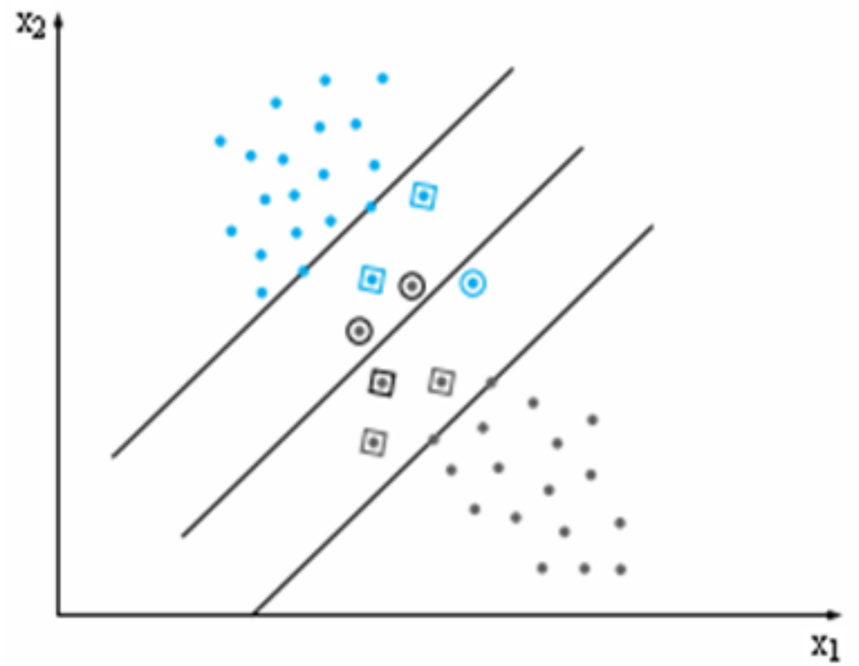
Cực tiểu hàm $J(\boldsymbol{w}) = \frac{\|w\|^2}{2}$

với ràng buộc $y^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) \geq 1 \ \forall t=1,\ldots,N.$

Các lớp không tách được tuyến tính

Dùng lề mềm, mong muốn:

- Các điểm ngoài miền lề được phân lớp đúng
- Lề cực đại nhưng ít lỗi nhất



-

Các lớp không tách được tuyến tính

Thêm biến chùng ξ đo mức lệch $\xi^t \geq 0$ từ $\boldsymbol{x}^t$ tới siêu phẳng lề:

Dẫn tới bài toán quy hoạch lồi sau để tìm **w** và w_0:

Cực tiểu hàm $J(\boldsymbol{w}, w_0, \xi) = \frac{\|w\|^2}{2} + C\sum_t \xi^t$

với các ràng buộc:

$y^t(\boldsymbol{w}^T\boldsymbol{x}^t + w_0) \geq 1 - \xi^t \quad \forall (\boldsymbol{x}^t, y^t) \in D,$

$\xi^t \geq 0 \qquad \forall$ t=1,...,N;

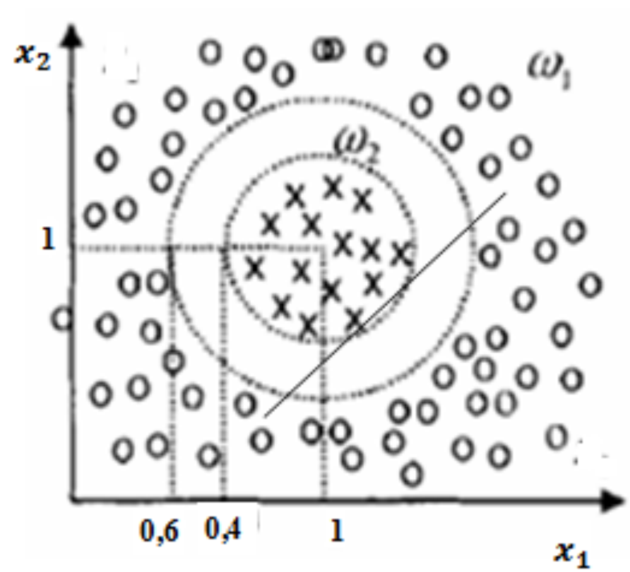
C là hằng số dương, phạt các điểm phân lớp sai,

trường hợp tách được tuyến tính ứng với $C \to \infty$.

$\Phi(x_1,x_2) = (z_1,z_2) = ((x_1 - 1)^2,(x_2 - 1)^2)$

- Nhiều bài toán, không dùng được phương pháp SVM nhưng khi nhúng các đối tượng vào không gian khác bằng một ánh xạ phi tuyến thì có thể áp dụng được
- Ví dụ. hai lớp trong hình sẽ tách được tuyến tính khi dùng phép biến đổi không gian đặc trưng :

$\Phi(\boldsymbol{x_1},\boldsymbol{x_2}) = \left((x_1-1)^2,(x_2-1)^2\right)$



Phương pháp hàm nhân:dùng hàm nhân để tìm hàm phân biệt

- Dùng ánh xạ $\Phi: X \to Z$ nhúng X lên gian Z (chiều lớn) để dễ xác định hàm phân biệt hơn.
- Xét tập mẫu $\{(\boldsymbol{x}^t, \boldsymbol{y}^t)\}_{t=1}^N$, $\boldsymbol{z}^t$ là ảnh của $\boldsymbol{x}^t$trong Z.
- Véctơ hệ số $\boldsymbol{w}$ của hàm *pbtt* trong Z là *thtt* của $\{\boldsymbol{z}^t\}_{t=1}^N$:

  $\boldsymbol{w} = \sum_t a_t \boldsymbol{z}^t = \sum_t \boldsymbol{a}_t \Phi(\boldsymbol{x}^t)$

  và hàm phân biệt: $g(\boldsymbol{x}) = \boldsymbol{w}' \Phi(\boldsymbol{x}) = \sum_t a_t \langle \Phi(\boldsymbol{x}^t), \Phi(\boldsymbol{x}) \rangle$.
- Để không phải biểu diễn Φ khi tính các tích trong $\langle \Phi(\boldsymbol{x}^t), \Phi(\boldsymbol{x}) \rangle$, ta dùng hàm hàm *nhân* $K(\boldsymbol{x}^t, \boldsymbol{x})$ sao cho: $K(\boldsymbol{x}^t, \boldsymbol{x}) = \langle \Phi(\boldsymbol{x}^t), \Phi(\boldsymbol{x}) \rangle \Rightarrow$ hàm phân biệt là:

  $g(\boldsymbol{x}) = \sum_t a_t \boldsymbol{K}(\boldsymbol{x}^t, \boldsymbol{x})$.

Các hàm nhân thông dụng

- *Đa thức bậc p*:

$$K(\boldsymbol{x}^t, \boldsymbol{x}) = (\boldsymbol{x}^T \boldsymbol{x}^t + 1)^2$$

trong đó $\boldsymbol{x}^T$ là ký hiệu vecto chuyển vị của $\boldsymbol{x}$.

Nói riêng, khi p=2 và không gian đặc trưng của $\boldsymbol{X}$ 2 chiều ta có:

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y} + 1)^2 (x_1 y_1 + x_2 y_2 + 1)^2$$

$$= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 y_1 x_2 y_2)^2 + x_1^2 y_1^2 + x_2^2 y_2^2,$$

tương ứng với $\Phi(\boldsymbol{x}) = (1, \sqrt{2}x_1,\ \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2.$

- *Hàm cơ sở bán kính* (*radial basis function*):

$$K(\boldsymbol{x}^t, \boldsymbol{x}) = exp\left[-\frac{\|\boldsymbol{x}^t - \boldsymbol{x}\|^2}{\sigma^2}\right].$$

Phân lớp Bayes

Xét bài toán *k* lớp

- Phân lớp thống kê: Hàm quyết định thường là $g_i(\boldsymbol{x})$= P(ω_i/$\boldsymbol{x}$),

  +Các xác suất được ước lượng từ DL quan sát được D,

  +Ví dụ. Fb gõ H cho kết quả hôm/ hôm nay/Hà Nội

- Phân lớp Bayes. Ước lượng được các xác suất P(ω_i) và P($\boldsymbol{x}$/ω_i), $i \leq k$

  +Dùng công thức Bayes để tính xác suất các hậu nghiệm P(ω_i/$\boldsymbol{x}$):

$$P(\omega_i / x) = \frac{P(x/\omega_i)P(\omega_i)}{P(x)} = \frac{P(x/\omega_i)P(\omega_i)}{\sum_{j=1}^{k} P(x/\omega_i)P(\omega_i)}$$

Quy tắc quyết định**MAP**: $x \in \omega_i \ if \ P(\omega_i / x) = \max\{P(\omega_j / x) : j \leq k\}$

Phân lớp Bayes: *Ví dụ về phân loại sản phẩm*

Bài toán: Có 2 loại sản: *tốt* (ω_1) và *thường* (ω_2); Tập mẫu: $n_1 = 1802840$ sp tốt ; $n_2 = 2704260$ sp thường; Một sp có đặc trưng $\boldsymbol{x}$: $P(\mathbf{x}/\omega_1) = 0{,}96$ và $P(\mathbf{x}/\omega_1) = 0{,}883$, cần quyết định sản phẩm này thuộc loại nào?

Giải. Ta có:

+$P(\omega_1) = n_1/n = 0{,}4$; $P(\omega_2) = n_2/n = 0{,}6.$

+ $P(\mathbf{x}/\omega_1)\, P(\omega_1) = 0{,}96.\ 0{,}4 = 0{,}384;$

+$P(\mathbf{x}/\omega_2)\, P(\omega_2) = 0{,}883.\ 0{,}6 = 0{,}49981$

Quyết định đối tượng là sản phẩm thường

Chuẩn hóa: $P(\omega_1/\boldsymbol{x}) = 0{,}3435$; $P(\omega_2/\boldsymbol{x}) = 0{,}5655$

Phân lớp Bayes

- *Quy tắc có khả năng/hợp lý nhất* (maximum likelihood: ML*):*

  +Khi các P(ω_i) như nhau thì QĐ $x \in \omega_i$ nếu P($\boldsymbol{x}$/ω_i) lớn nhất.

  +Trong ví dụ ***x*** là SP tốt: $P(\boldsymbol{x}/\ \omega_1)=0,96$ *và* $P(\boldsymbol{x}/\ \omega_1)= 0,883$

- *Quy tắc cực tiểu rủi ro*.

  Khi ta quyết định $x \in \omega_i$ → có hành đông α_i nhưng nếu nó thuộc ω_j thì thiệt hại $\lambda(\alpha_i/\omega_j)$. Thiệt hại trung bình sẽ là:

$$R(\alpha_i / x) = \sum_{j=1}^{k} \lambda(\alpha_i/\omega_\mathrm{j})\mathrm{p}(\omega_\mathrm{j} / x)$$

Quy tắc quyết định là chọn $x \in \omega_i$ có R(α_i/***x***) nhỏ nhất

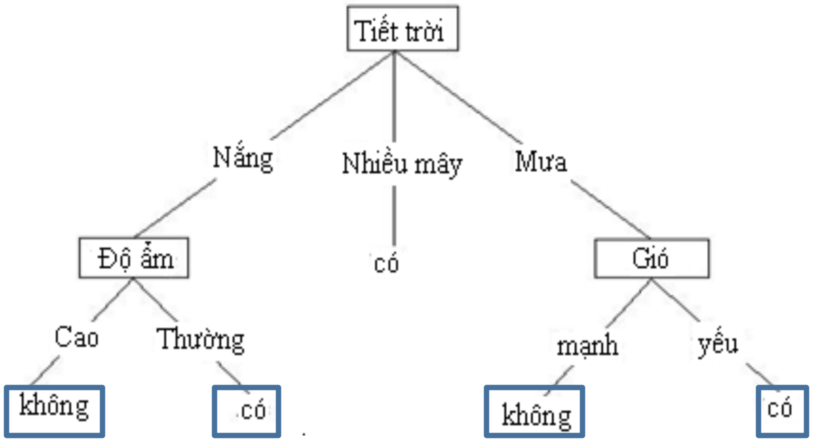
Ví dụ. nếu SP ***x*** tốt bị xếp SP thường mất 3$ (lệch giá), còn thường xếp tốt sẽ bị khiếu nại và mất 2$ thì R(α_1/***x***) =2x0,56550,435=0.87;

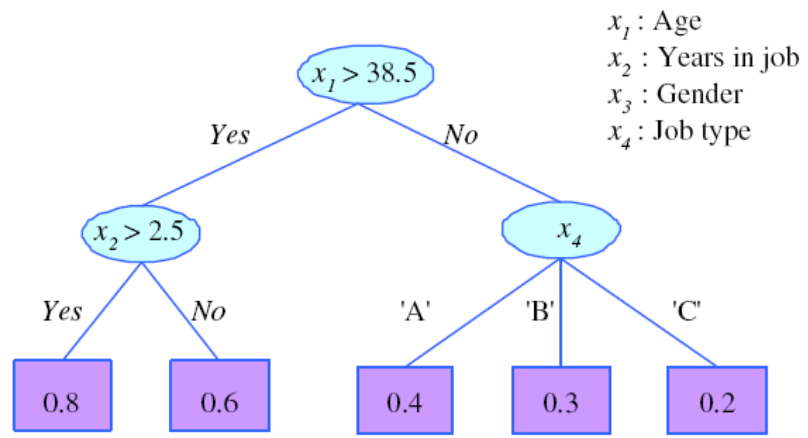
R(α_2/***x***) =3x0,565=1.695. **Ta chọn *x* là SP tốt**

Cây quyết định

Mỗi cây quyết định cho môt tâp quy tắc phân lớp hoặc hồi quy:

- Mỗi đường di từ gốc đến là cho 1 luật
- Mỗi nút ≠ lá biểu thị một *thuộc tính/điều kiện* kiểm tra
- Mỗi cạnh biểu thị *giá trị kiểm tra* của thuộc tính tương ứng
- Mỗi lá cho một *giá trị nhãn* xác định bởi luật





Nếu Tiết trời = nắng $\wedge$ *độ ẩm = cao thì* $h(x)$ *= không*

Nếu $(x_1 > 38{,}5) \wedge (x_2 > 2{,}5)$ *thì* $h(x) = 0{,}8$

Mô tả chung các thuật toán ID3, C45, CART

- Để đơn giản: xét bài toán 2 lớp với nhãn dương tính + /âm tính -
  các thuộc tính nhận giá trị trong tập rời rạc, hữu hạn, tập DL quan sát D
- Quá trình xây dựng cây của các thuật toán thực hiện từ gốc đến lá:
  + Mỗi thuật toán có một tiêu chuẩn chọn thuộc tính tốt nhất
  +Mở một nút gốc,
    - nếu S có nhãn thuần nhất thì gán nhãn cho nút và dừng,
    - nếu tập thuộc tính rỗng thì gán nhãn cho nút theo đa số và dừng
    - Còn lại, thuộc tính tốt nhất là nhãn cho nút, tạo các nhánh từ nút ứng với các giá trị thuộc tính được chọn
    - Tạo nút gốc cho mỗi nhánh với tập mẫu là tâp con của S có giá trị phù hợp với cạnh tương ứng
    -Lặp lại quá trình trên cho các tập mẫu mới đến khi kết thúc
- Lược đồ chung của C45 và CART giống với ID3 (Quinlan 1986)

Thuật toán ID3 xây dựng cây (*Tập mẫu, tập nhãn, tập thuộc tính*)

Bước 0. Khởi tạo: D, tập nhãn, tập thuộc tính;
Bước 1. Tạo nút gốc (Root) cho cây;
Bước 2. // Gán nhãn cho nút nếu dữ liệu thuần nhất hoặc tập thuộc tính là rỗng.
    2.1. Nếu mọi mẫu đều dương tính thì nhãn nút gốc = +
    2.2. Nếu tất cả các mẫu là âm tính thì nhãn nút gốc = -
    2.3. Nếu tập các thuộc tính là rỗng trả về cây một nút gốc có nhãn = *giá trị phổ biến nhất* của thuộc tính đích trong tập các mẫu;
    2.4 Các trường hợp khác sang bước 3;
Bước 3.
    3.1. *Xác định thuộc tính* phân loại tập mẫu tốt nhất trong tập thuộc tính;
    3.2. A ← thuộc tính phân lớp tốt nhất;
    3.3. Với mỗi giá trị có thể v_i của thuộc tính A, thực hiện:
        3.3.1. Thêm một nhánh mới dưới nút gốc với mỗi điều kiện $A=v_i$;
        3.3.2. Xác định $Examples_v$ = {**x** ∈ *tập mẫu*: **x** có giá trị v_i ở thuộc tính *A*};
        3.3.3. Nếu Examples_v là rỗng thì thêm dưới nhánh một nút lá có nhãn là nhãn phổ biến nhất của các mẫu trong *tập mẫu;*
        3.3.4. Ngược lại, trở lại bước 1 với khởi tạo:
            (D = $Examples_v$, *tập nhãn, tập thuộc tính – {A})*

Tiêu chuẩn thuộc tính tốt nhất

Mỗi tập S có k nhãn, ta ký hiệu S^i là tập con có nhãn c_i, $p_i = |S^i| / |S|$,
V_A là tập gián trị của thuộc tính A tương ứng, $\forall v \in V_A$, S_v= tập nhận giá trị v ở tt A

- ID3 dùng *thu hoạch thông tin* (Information Gain) cực đại để chọn thuộc tính

  + khái niệm entropy: $Entropy(S) = -\sum_{j=1}^{k} p_i \log p_i$

  + $\mathrm{Gain(S, A) = Entropy(S)} - \sum_{\mathrm{v \in V_A}} \frac{|\mathrm{S_v}|}{|\mathrm{S}|} Entropy(s_v)$

- Với A có V_A lớn thì $Gain(S.A)$ thường lớn, C45 dung***Tỷ lệ thu hoạch thông tin*** (*GainRatio (S,A*))cho các trường hợp này:

  + $\mathrm{SplitInformation(S, A)} = \sum_{\mathrm{v \in S_A}} \left(|\mathrm{S_v}| / |S|\right) \log\left(|\mathrm{S_v}| / |S|\right)$

  + $\mathrm{GainRatio\ (S, A) = Gain(S, A)/SplitInformation(S, A)}$

- CART dùng thu hoach Gini, định nghĩa dựa trên độ thuần nhất Gini: GI(S) (xem trang sau)

Khác biệt giữa ID3 với C45 và CART

- Tiêu chuẩn chọn thuộc tính tốt nhất

Các điểm khác nhau khác:

- C45 và các biến thể:

  + Khắc phục phù hợp trội (overfititfing): Chặn sớm, tỉa cây, lược sau luật

  +Xử lý dữ liệu vị mất

  + Xử lý thuộc tính thực: rời rạc hóa hoặc rẽ nhánh

- CART chú trọng xử lý thuộc tính thực và bài toán hồi quy.

  + $GI(S) = 1 - \sum_{j=1}^{k} p_i^2(S)$

  + Với mỗi tt A, điểm chia x_A^c S được chia thành S_L và S_R (trái và phải)

  $+ \Delta GI(S, A, x_A^c) = GI(S) - \left[p_L GI(S_L) + P_R GI(S_R)\right]$

  $+ \Delta GI(S, A) = \max\left\{\Delta GI(S, A, x_A^c)\right\}$

Rừng ngẫu nhiên (Random Forest)

- Khi dữ liệu chiều cao, xây dựng cây lâu và chất lượng nhận dạng (Phân lớp /hồi quy) thấp.

  Người ta dùng rừng ngẫu nhiên
- Rừng ngẫu nhiên là một bộ nhận dạng bao gồm một tập bộ phân lớp cơ sở dạng cây quyết định được kết hợp theo phương thức bỏ phiếu.
- Các bộ cơ sở được xây dựng từ các tập DL con với tập con đặc trưng khác nhau được lấy ngẫu nhiên từ tập DL quan sát được.

Rừng ngẫu nhiên

Thủ tục xây dựng rừng ngẫu nhiên gồm ba pha:

- tạo dữ liệu (tạo vectơ ngẫu nhiên),
- xây dựng các cây cơ sở,
- kết hợp các cây cơ sở theo phương thức bỏ phiếu.

Pha tạo dữ liệu. $\mathrm{D} = \{(\boldsymbol{x}^k, y^k)\}_{k=1}^{N}, \boldsymbol{x}^k \in \mathcal{R}^n$

Chọn M ($< N$) và m ($< n$).

Để có $\mathrm{D_k}(\subset \mathcal{R}^m$ cho CQĐ $\mathrm{T_k}$ ($k \leq L$),

chọn NN m đặc trưng, lấy ngẫu nhiên M đối tượng từ D rồi chiếu nó lên các đặc trưng được chọn này.

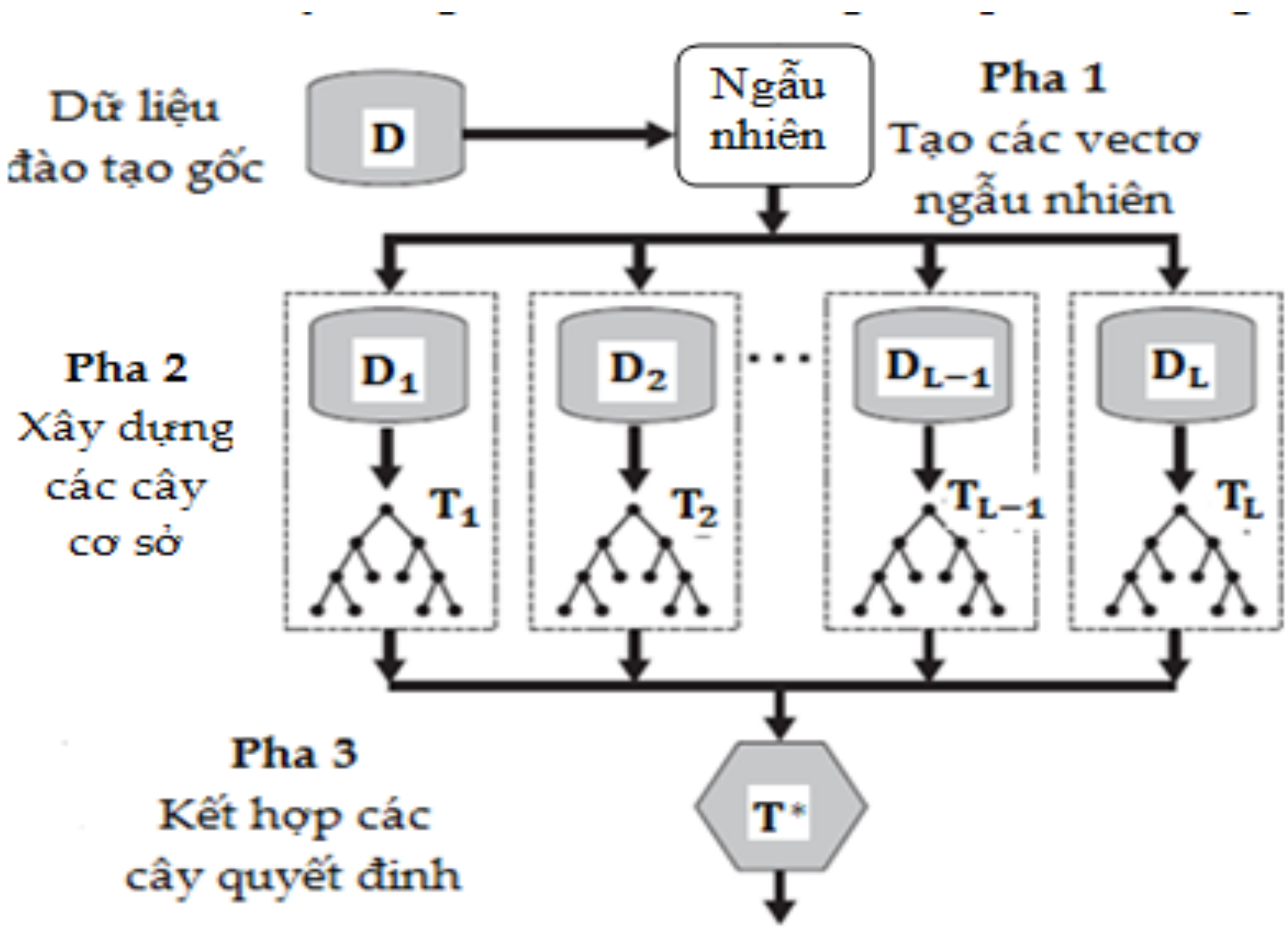
Rừng ngẫu nhiên

Chọn số đặc trưng m

Breiman gợi ý chọn m như sau:

- Đối với bài toán phân lớp, $m = \lfloor\sqrt{n}\rfloor$, trong đó $\lfloor a \rfloor$ ký hiệu phần nguyên của a.
- Đối với bài toán hồi quy, $m = \lfloor n/3 \rfloor$

Rừng ngẫu nhiên



Thuật toán xây dựng rừng ngẫu nhiên

Bước 1. Với $k=1$, đến L thực hiện:

1.1. Lấy ngẫu nhiên m đặc trưng $\{A_{i1}^{k}, \ldots, A_{im}^{k}\}$ của D;

1.2.Lấy ngẫu nhiên tập R_k gồmM dữ liệu trong D ;

1.3. D_k = Hình chiếu của R_k lên các đặc trưng $\{A_{i1}^{k}, \ldots, A_{im}^{k}\}$;

1.4. Xây dựng cây T_k từ tập D_k; có bộ nhận dạng c_k

Bước 2. Kết hợp bỏ phiếu trọn số đều $\{C_k\}_{k=1}^{L}$

Đầu ra của hệ cho đối tượng x sẽ là:

- Đối với bài toán hồi quy: $C(x) = \frac{1}{L}\sum_{k=1}^{L} C_k(x)$
- Đối với bài toán phân lớp: $C(x) = \left[\frac{1}{L}\sum_{k=1}^{L} C_k(x)\right]$,

Tìm hàm hồi quy

Bài toán. Hàm chưa biết f: $\boldsymbol{X} \rightarrow R^n$ có tập DLquan sát được:

D= $\left\{(x^k, y^k)\right\}_{k=1}^{N}; x^k \in X, y^k (\in R^n) \approx f(x^k)\ \forall k$

*V*ới hàm $\varphi(\boldsymbol{x},\boldsymbol{c})$, trong đó $\boldsymbol{c} = (c_1, \ldots, c_m)^T$, M ≤ N, là vectơ tham số cần tìm để hàm hồi quy g($\boldsymbol{x}$) = $\varphi(x, c_1, \ldots, c_M)$ cực tiểu sai số trung bình phương (SSE):

$$E(\mathbf{c}) = \sum_{j=1}^{N} d^2(g(x^j), y^j),$$

($d(\boldsymbol{z},\boldsymbol{y})$ là khoảng cách Euclid của $\boldsymbol{z}$ và $\boldsymbol{y}$ trong R^n)

- Để đơn giản, ta xét $n = 1$, $E(c) = \sum_{j=1}^{N} \left[\varphi(x^j, c_1, \ldots, c_M) - y^k\right]^2$
- Thường dùng phương pháp gradient để tìm cực tiểu E
- *H*àm $\varphi(\boldsymbol{x},\boldsymbol{c})$ dạng tuyến tính: $\varphi(x,c) = \sum_{k=1}^{M} c_k \varphi_k(x)$

Trong đó $\{\varphi_k\}_{k \le M}$ là hệ hàm độc lập tuyến tính cho trước

Thuật toán gradient

Bước 1. Khởi tạo và vecto $c = c^0 \in R^M$ tùy ý;

Bước 2. Thực hiện lặp:

2.1. Tính $E'(c^0) = \left[\frac{\partial E}{\partial c_1}, \dots, \frac{\partial E}{\partial c_1}\right]^T$; // chỉ số trên ^{T} là vecto chuyển vị

2.2. $\alpha_1 = \alpha$ // $\alpha \in (0,1]$ cho trước

2.3. $c^1 = c^0 - \alpha_1 E'(c^0)$;

2.3. Nếu $E(c^1) < E(c^0)$ thì $c^0 \leftarrow c^1$ và quay lại 2.1;

2.4. ngược lại: $\alpha_1 \leftarrow \frac{\alpha_1}{2}$ và trở lại 2.3;

Thuật toán dừng khi $|E'(c^1)|$ đủ bé

Trường hợp tuyến tính

Khi $\varphi(x,c)$ có dạng tuyến tính:

$$\varphi(x,c) = \sum_{k=1}^{N} c_k \varphi_k(x)$$

Cực tiểu tại điểm dừng của E: cho bởi $\frac{\partial E}{\partial c_i}=0;\ i=1,\ldots,M$

$$\frac{\partial E}{\partial c_i} = 2\sum_{k=1}^{N}\left[\sum_{j=i}^{M} c_j \varphi_j(x^k) - y^k\right]\varphi_i(x^k) = 0$$

$$\Leftrightarrow \sum_{k=1}^{N}\left[\sum_{j=i}^{M} c_j \varphi_j(x^k) - y^k\right]\varphi_i(x^k) = 0\ \forall i$$

$$\Leftrightarrow \sum_{j=1}^{M} c_j\left[\sum_{k=i}^{N} \varphi_j(x^k)\varphi_i(x^k)\right] = \sum_{k=1}^{N} y^k \varphi_i(x^k)\quad \forall i.$$

Có thể giải trực tiếp hệ phương trình này khi M không lớn

Tìm hàm hồi quy: Vì sao cần M $\leq$ N?

- Khi số M > N thì thường có vô số nghiệm $\boldsymbol{c}$ để E(c)= 0, hàm hồi quy khống đáng tin.

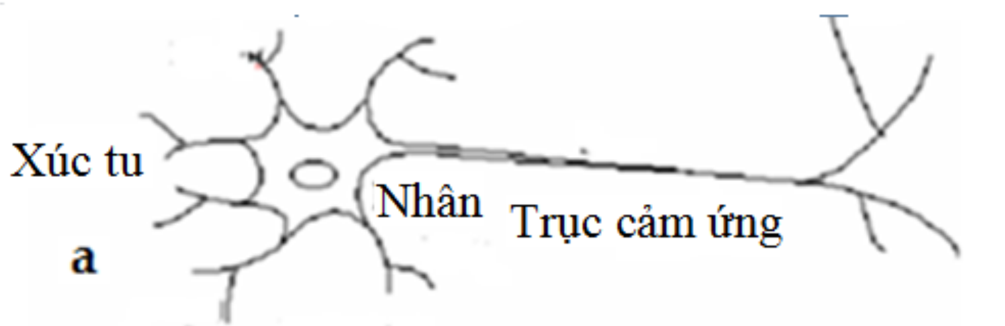
  Ví dụ: khi X= Y = R, N=2, M $\geq$ 3 thì có vô số đường bậc 3 qua 2 điểm

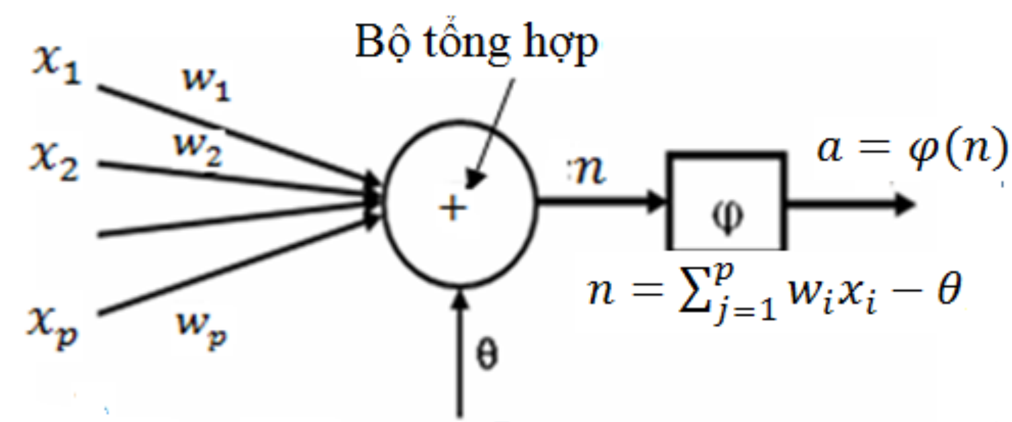
- Khi M =N thường tìm được duy nhất/hữu hạn c để $E(\boldsymbol{c}) = 0$:

  Tức là $g(\boldsymbol{x}^k)=\boldsymbol{y}^k$ với mọi k

Mạng nơron

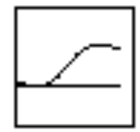
- Mô hình nơron: nơron tự nhiên và nơron nhân tạo





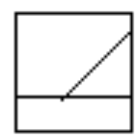
Một số hàm chuyển thông dụng

- **Hàm log_sig :** (log-sigmoid)
  - a= logsig(n)= $\dfrac{1}{1+e^{-n}}$ Ký hiệu:

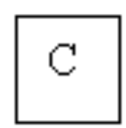


- **Hàm tanghyperbolic :** (tansig)
  - a= tansig(n)= $\dfrac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$ Ký hiệu:

- **Hàm tuyến tính dương:** (poslin)
  - a= poslin(n)= $\begin{cases} n & n \geq 0 \\ 0 & n < 0 \end{cases}$ Ký hiệu:

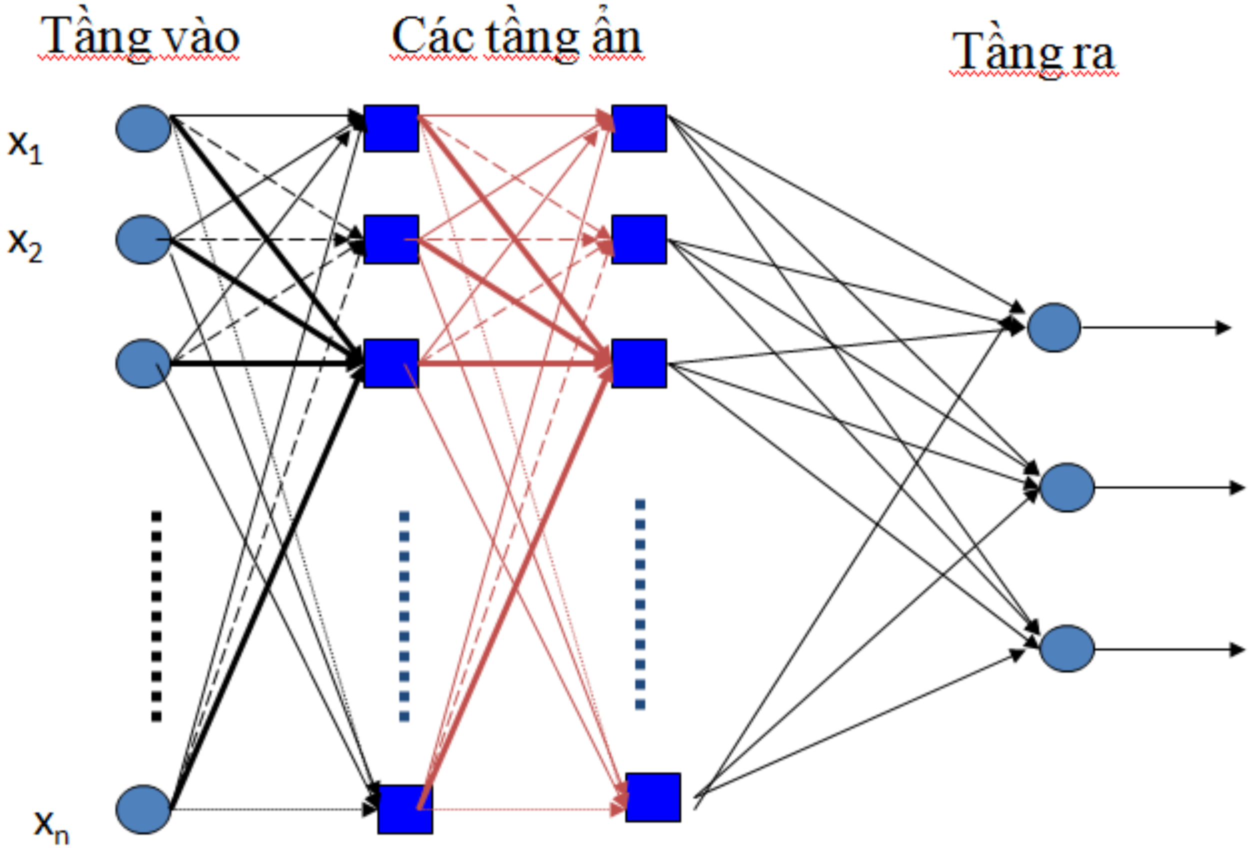


- **Hàm cạnh tranh:** (compet)
  - a= compet(n)= $\begin{cases} 1 & n = \max \\ 0 & n < \max \end{cases}$ Ký hiệu:



(Compet)

Kiến trúc mạng MLP (MultiLayer Perceptron)



Mạng $R - S^1 - S^2 - S^3$

Đặc điểm kiến trúc và thuật toán huấn luyện mạng MLP:

- Đặc điểm kiến trúc:

  + Tầng vào vào có m+1 nút nếu $X \subset R^m$

  + Tầng ra n nơron nếu $Y \subset R^n$

  +***Một tầng ẩn với số nơron thích hợp và DL quan sát được đủ nhiều thì xấp xỉ được hàm liên tục với sai số đủ nhỏ***

  + ***Kiến trúc tối ưu*** còn là bài toán mở

- Dùng thuật toán gradient ngẫu nhiên (*Thuật toán lan truyền ngược:BP*) :

  1-Khởi tạo c^0 tùy ý

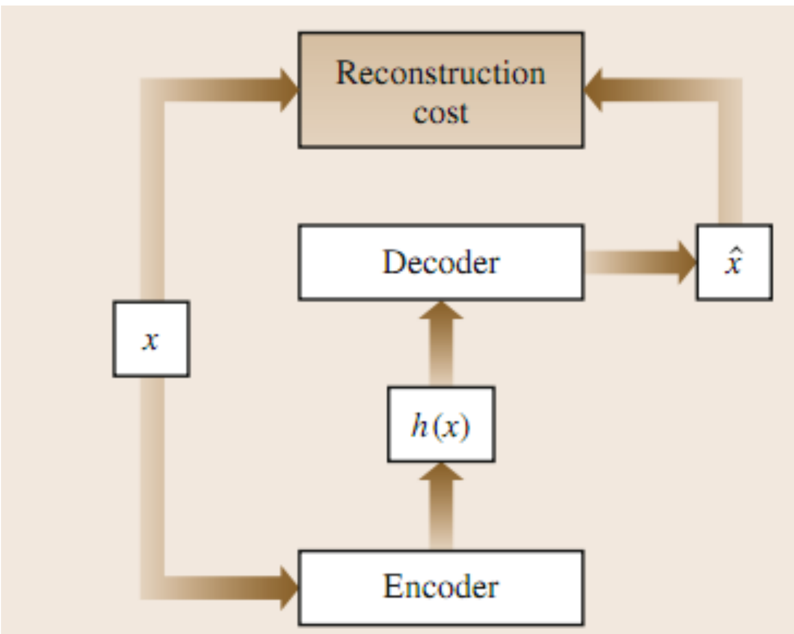
  2- Thực hiện lặp:

  2.1- Mỗi lần lấy ngẫu nhiên một xk làm đầu vào mạng để tính đầu ra và sai số E

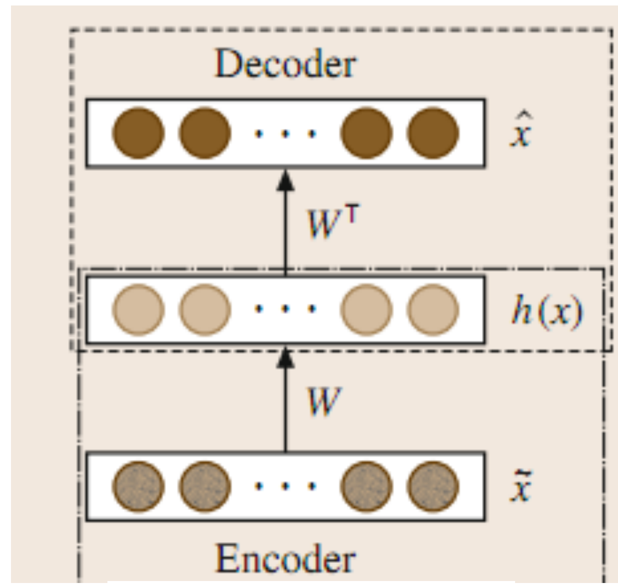
  2.2- Hiệu chỉnh tham số tuần từ tầng ra đến tầng ẩn đầu tiên

Học sâu

- Dùng mạng MLP, kiến trúc và huấn luyện lai
- Khối tự động mã hóa: Khử nhiễu



$$\hat{x} = f(Wh(x) + b_0)$$

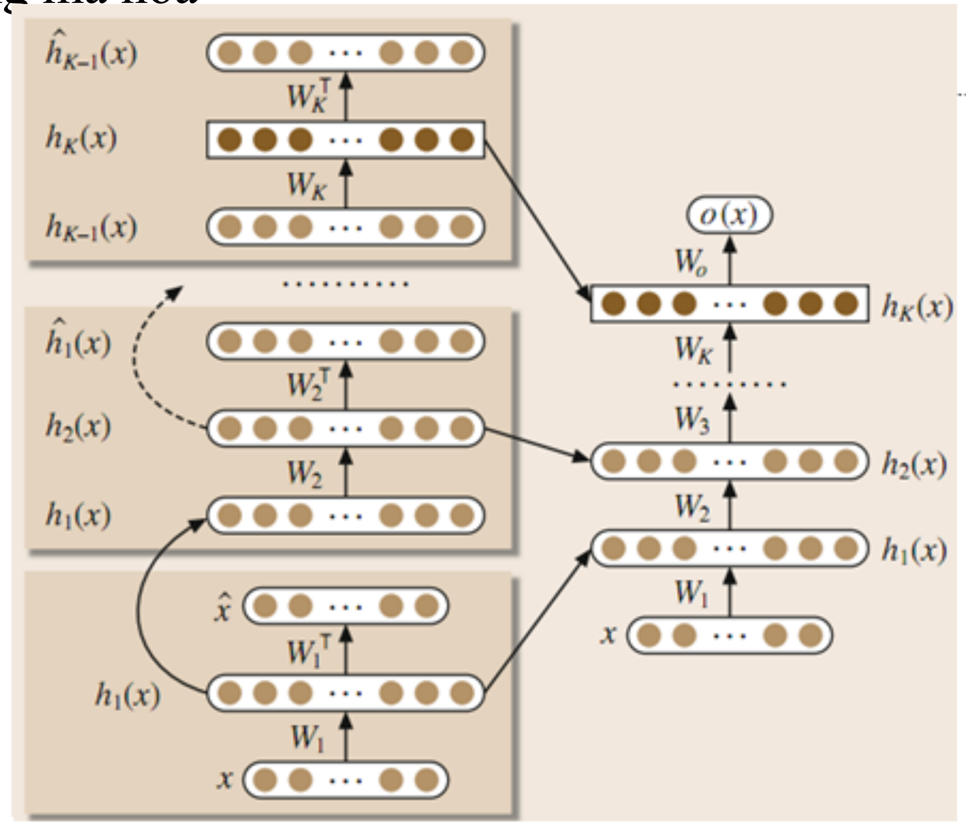


$$\boldsymbol{h}(\tilde{\boldsymbol{x}}) = f(W\tilde{\boldsymbol{x}} + \boldsymbol{b}_h)$$

$$\hat{\boldsymbol{x}} = f(W^T\boldsymbol{h}(\tilde{\boldsymbol{x}}) + \boldsymbol{b}_o)$$

Học sâu

Học sâu tự động mã hóa



Tài liệu tham khảo

1. Hoàng Xuân Huấn, *Giáo trình Học Máy*, NXB ĐHQG HN, 2015
2. J. Kacprzyk, W. Pedrycz (Editors.), *Handbook of Computational Intelligence*, Springer, 2015

Cám ơn đã chú ý lắng nghe!