بازشناسی الگو

جلسه هفتم

طراحی طبقه بندی کننده (کلاسیفایر)

کلاسیفایر های خطی

فهرست مطالب

- مقدمه و جایگاه بحث
- طراحی کلاسیفایر در بازشناسی الگو
- معمولترین روش های طرح کلاسیفایر
- سیستم های خطی کلاسیفایرو مزیت های آن
- معرفی مبانی طراحی کلاسیفایر خطی
- تعمیم توابع جدا ساز خطی
- خلاصه و نتیجه گیری از بحث

بازشناسی آماری الگو

- مباحث اصلی درSPR:
  - طراحی کلاسیفایر
    - با داشتن توزیع نمونه ها در کلاس ها(کلاسیفایر Bayes و توابع جدا ساز)
    - بدون داشتن اطلاعات توزیع کلاس ها
      - یافتن توزیع و استفاده از روش های قبلی
      - طراحی کلاسیفایر به صورت مستقیم
  - ارزیابی کارآیی (تحلیل خطا و ریسک)
    - با استفاده از شمارش
    - یافتن حد قابلیت تعمیم
  - بازنمایی نمونه ها(استخراج و انتخاب مشخصه ...)

جنبه های عملی

- دیدیم که در عمل از آنجائیکه تنها تعدادی نمونه در اختیار داریم، مسئله به این شکل در می آید که به چه نحو، از روی این نمونه ها، کار های فوق را انجام دهیم.
- مسائل فرعی برای هر یک ازمنظور های فوق:
  - تخمین احتمال یا توزیع
  - اجرای فرآیند های بهینه سازی

منظور از مسئله طرح کلاسیفایر:

- با بازنمایی نمونه ها(بردار مشخصه ها) به نحوی آنها را دستکاری کنیم که تصمیم گیری در خصوص تفکیک آنها میسر شود:
  - نوع دستکاری داده ها برای یافتن کمیت معیار جدا سازی (محاسبات نرم)
  - نوع تصمیم گیری (عملیات سخت)
- قبلا دیدیم که یک مسیر برای انجام این کار، یافتن توزیع داده های هر کلاس از روی نمونه های یادگیری آن کلاس و استفاده از قانون Bayes است.
- آنچه در این چند جلسه طرح می شود، استفاده از شیوه های مستقیم طراحی کلاسیفایر از روی نمونه های یادگیری است.

صورت مسئله طراحی مستقیم کلاسیفایر

- در موارد عملی، عموما با این مسئله مواجه هستیم که با داشتن تعدادی نمونه (مثال) قرار است سیستمی طرح شود که بتواند آنها را از هم جدا نماید.
- خود این مسئله می تواند شکل با مربی یا بدون مربی داشته باشد. در این بحث توجه ما به شکل با مربی متمرکز شده است.
- سه ملاحظه اصلی:
  - بیشترین توفیق در تفکیک نمونه های داده شده(یادگیری)
  - بیشترین توفیق در نمونه های دیده نشده(تعمیم)
  - عدم مواجهه با مشکلات عملی محاسباتی

مهمترین روش های طراحی مستقیم کلاسیفایر

- روش جدا سازی(کلاسیفایر) خطی
- روش نزدیکترین همسایگی (NN KNN)
- روش درخت تصمیم
- روش بردار های پشتیبان SVM
- روش شبکه های عصبی

مزیت های کلاسیفایر خطی

- سادگی در تحلیل تئوری
- سادگی در برآورد کارآیی
- وجود (و جا افتادگی)ابزار های ریاضی (نرم افزار، الگوریتم ها ...)
- قابلیت انتقال مطالب به ادبیات دیگر زمینه ها

  تخمین و شناسایی، تحلیل پایداری، ....
- تجزیه و تحلیل های این روش مبنای اولیه بسیاری روش های دیگر بوده است.

یادآوری: توابع جدا ساز
Discriminant functions

- این توابع میدان های اسکالر هستند(ورودی بردار و خروجی اسکالر) که بر حسب کمیت خروجی آنها، می توان به تصمیم گیری در مورد انتساب مبادرت نمود.
- در فضای مشخصه ها، می توان منحنی (پوسته) ای (مرز تصمیم) پیدا نمود که در دو طرف آن، نتیجه مقایسه دو تابع جدا ساز، متفاوت بوده و بتوان دو ناحیه مختلف را که هر یک منتسب به یکی از کلاس ها هستند، تفکیک نماید.
- یکی از دلخواه ترین حالت ها آن است که این توابع، خطی باشند.

توابع جدا ساز خطی و توزیع

- دیدیم که در شکل های خاصی ازتوزیع نمونه ها، (توزیع نرمال) این توابع، خطی بوده کاملا معادل کلاسیفایر Bayesian دارند.
- ضرایب تعیین کننده خط مزبور ارتباط معنا داری با پارامتر های تابع توزیع داشت.
- در حالتی که توزیع، غیر نرمال باشد، توابع جدا ساز (بدست آمده از این روش) خطی نخواهند بود ولی می توان از شیوه های دیگری برای طرح کلاسیفایر خطی استفاده کرد.

صورت مسئله طراحی کلاسیفایر خطی:

- دیدیم در شکل عملی مسئله، با حالتی مواجه هستیم که تنها تعدادی مثال یا نمونه در دسترس است و طراحی کلاسیفایر با استفاده از آنها باید صورت گیرد.
  - یافتن ضرایب خطی که بتواند فضا را به دو بخش، که هر یک منتسب به یکی از کلاس ها هستند، تفکیک نماید.
- رویکرد های اصلی:
  - یافتن توزیع از روی نمونه هاو یافتن ضرایب خط LDF از روی پارامتر های تابع توزیع
  - یافتن مستقیم ضرایب خط از روی داده ها

چند نکته

- در این بحث تا حد زیادی می توان شکل کار را به شکل مسئله یافتن تابع توزیع شبیه دانست.
- ما دنبال تابعی با شکل شناخته شده و پارامتر های مجهول هستیم.
- هدف، پیدا کردن بهترین خط است. (یک مسئله بهینه سازی)
- با توجه به خصوصیات مسئله (گسسته بودن کلاسیفایر) بدنبال راه حل های نزدیک بهینه (و نه لزوما بهینه) خواهیم بود.
- بیشتر تحلیل های اولیه، به مسئله دو کلاسه اشاره می کند و سپس روش هایی برای تعمیم آن به مسائل چند کلاسه پیشنهاد می شود.

Linear discriminant functions

- تابعی شامل ترکیب خطی از مولفه های مختلف ورودی (**X**):

$$g(x) = w^t x + w_0 \qquad (1)$$

که **w**مشخص کننده جهت خط و **w0** بایاس یا عرض از مبدا آن است.

- مبنای تصمیم گیری:

$$\omega_1 \;;\; \text{if } g(x) > 0 \quad \text{and } \omega_2 \;;\; \text{if } g(x) < 0$$

$$\omega_1 \;;\; \text{if } w^t x > -w_0 \quad \text{and } \omega_2 \;;\; \text{otherwise}$$

حالت $g(x) = 0$، *را که در اصل هدف نهایی است(مرز تصمیم)، عملا در بسیاری از موارد عملی قابل یافتن نیست.(گسسته بودن مسئله).*

سطح یا رویه تصمیم

- $g(x) = 0$ در واقع سطح تصمیم گیری (جدا کننده دو زیر فضای متعلق به کلاس ها) را تعیین می کند
- برای حالتی که تابع جدا ساز خطی است، این سطح یک صفحه چند بعدی hyperplane است.
- همانطوریکه می دانیم برای یافتن فاصله یک نقطه تا یک رویه (صفحه) می توان مختصات آن نقطه را در معادله صفحه قرار داد زیرا اصولا:
  - فاصله نقطه از صفحه در واقع فاصله آن نقطه با نقطه مشترک بین "خط عمود بر صفحه و گذرنده از نقطه اول" و " صفحه" است.
  - جهت هر صفحه، با بردار عمود بر آن صفحه مشخص می گردد.
- این مسئله جذابیت ویژه ای به کلاسیفایر های خطی می بخشد.

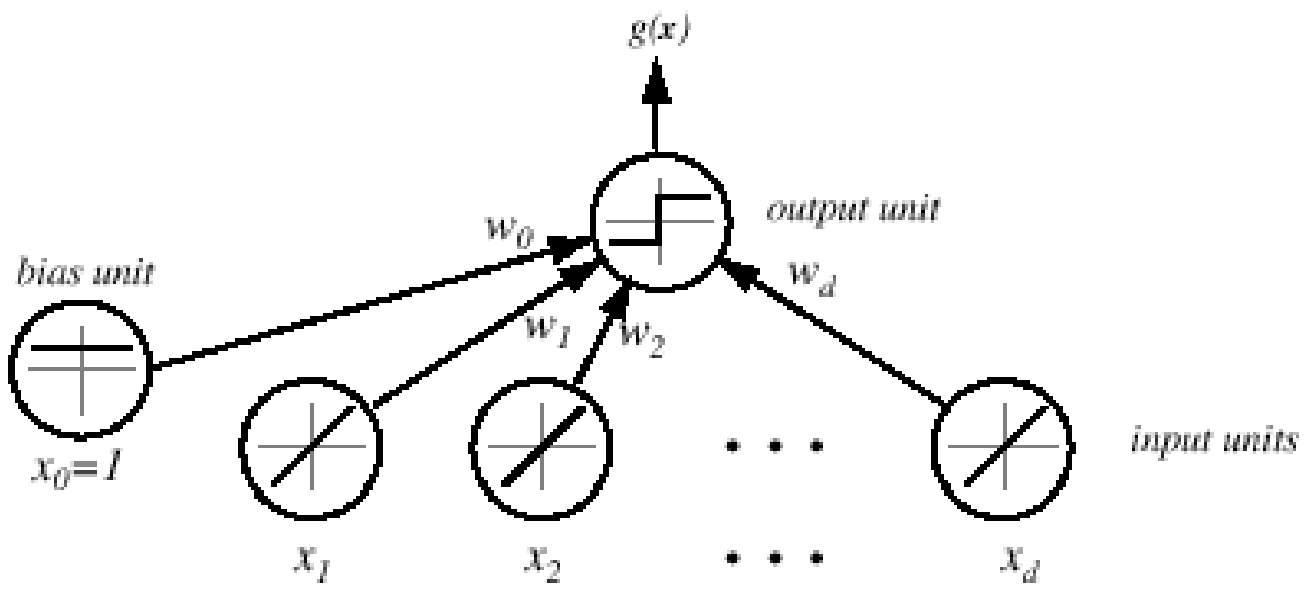

FIGURE 5.1. A simple linear classifier having d input units, each corresponding to the values of the components of an input vector. Each input feature value x_i is multiplied by its corresponding weight w_i; the effective input at the output unit is the sum all these products, $\sum w_i x_i$. We show in each unit its effective input-output function. Thus each of the d input units is linear, emitting exactly the value of its corresponding feature value. The single bias unit unit always emits the constant value 1.0. The single output unit emits a +1 if $\mathbf{w}^t\mathbf{x} + w_0 > 0$ or a −1 otherwise. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

نمايش ترسيمی برای مرز تصميم

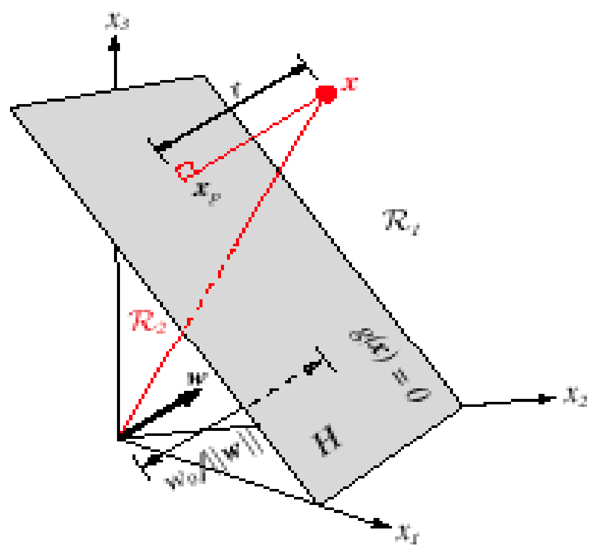

FIGURE 5.2. The linear decision boundary H, where $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x}+w_0 = 0$, separates the feature space into two half-spaces $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

خصوصیات خط

می دانیم که در فضای چند بعدی، یک خط(صفحه) را می توان به بردار یکه حامل آن و یک نقطه از آن نشان داد

$$x = x_p + \frac{r.w}{\|w\|} \quad \text{(since w, is colinear with x - x}_p \text{ and } \frac{\text{w}}{\|\text{w}\|} = 1)$$

$$\text{since } g(x) = 0 \text{ and } \quad w^t.w = \|w\|^2; \quad \textit{therefore} \quad r = \frac{g(x)}{\|w\|}$$

$$\textit{in particular}: \ d(0,H) = \frac{w_0}{\|w\|}$$

حالت چند کلاسه :

$$g_i(x) = w_i^t x + w_{i0} \qquad i = 1,\ldots, c$$

- در این حالت ما به تبیین C خط مختلف می پردازیم:

assign x to ω_i if $g_i(x) > g_j(x) \ \forall j \neq i$,

- این کلاسیفایر را یک ماشین خطی می گویند
- ماشین خطی فضا ی مشخصه ها را به C ناحیه تقسیم می کند.
- در هر ناحیه، مقدار تابع خطی مربوط به یکی از کلاس ها (مثلا کلاس i ام که صاحب آن ناحیه است) از بقیه بزرگتر است.

$$d(x, H_{ij}) = \frac{g_i - g_j}{\left\| w_i - w_j \right\|}$$

- لذا اگر نمونه تست در این ناحیه قرار گرفته باشد، ان را به کلاس i منسوب می کنیم.

حالت دو کلاسه

- برای مسئله دو کلاسه تعیین مرز ساده است، می توان خط (یا ابر صفحه) جدا کننده را به نحو زیر تعریف کرد:
  - hyperplane :

$$g_i(x) = g_j(x)$$

$$\Leftrightarrow (w_i - w_j)^t x + (w_{i0} - w_{j0}) = 0$$

  - $w_i - w_j$ is normal to H_{ij} and

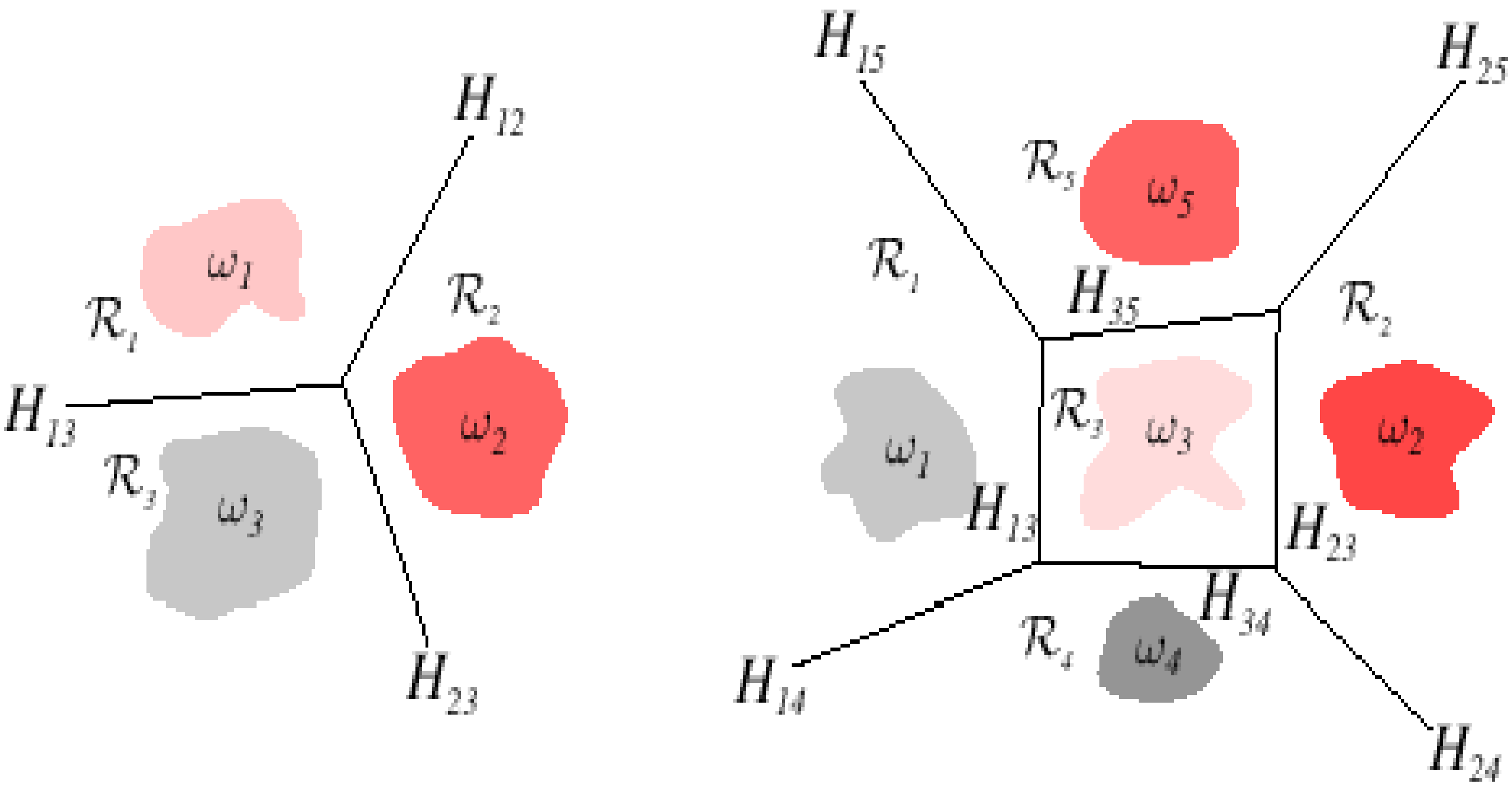

FIGURE 5.4. Decision boundaries produced by a linear machine for a three-class problem and a five-class problem. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

طراحی کلاسیفایر خطی بر اساس نمونه های موجود حالت دو کلاسه

- برای یک مسئله دو کلاسه، صورت مسئله طراحی یک کلاسیفایر خطی اینچنین است:
  - با داشتن تعدادی نمونه از دو کلاس C1 و C2 که در دو مجموعه H1 و H2 گرد آوری شده اند:
  - هدف پیدا کردن W و W0 به گونه ای است که بتواند این دو مجموعه را کاملا از هم جدا سازد بطوریکه برای هر نمونه:

$$W_{12}x_i - W_0 = \begin{cases} >0 & \mathit{for} \quad x_i \in H_1 \\ =0 & \\ <0 & \text{for} \quad x_i \notin H_2 \end{cases}$$

پیاده سازی عملی

- اگر x دارای d بعد باشد، به d+1 پارامتر نیازمندیم
- معمولا برای راحتی در عملیات جبر خطی ابتدا مسئله را به فضای d+1 بعدی می بریم تا همزمان بردار وزن و بایاس با هم محاسبه شوند

$$W = \begin{pmatrix} w_{12} \\ w_0 \end{pmatrix} \quad ; \quad \hat{x}_i = \begin{pmatrix} x_i \\ 1 \end{pmatrix}$$

so

$$\hat{x}_i . W \begin{cases} > 0 & \text{if } \hat{x}_i \in H_1 \\ < 0 & \text{if } x_i \in H_2 \end{cases}$$

پیاده سازی عملی

- اگر نمونه های یکی از کلاس ها (مثلا H1 تبدیل بعد یافته) را در نظر بگیریم(A)، باید برای همه آنها داشته باشیم:
- در عین حال برای نمونه های کلاس دوم اگر آنها را در ماتریس B قرار داده باشیم:
- حال:
  - ابتدا مسئله تبدیل نامعادله به معادله را داریم
  - سپس حل معادله با استفاده از گرفتن معکوس ماتریس

$$A = \begin{bmatrix} \hat{x}_1^T \\ \hat{x}_2^T \\ . \\ . \\ \hat{x}_n^T \end{bmatrix} ; \qquad A.W > 0$$

$$B = \begin{bmatrix} \hat{x}_1^T \\ \hat{x}_2^T \\ . \\ . \\ \hat{x}_n^T \end{bmatrix} ; \qquad B.W < 0$$

Linear seprability
جدایی پذیری خطی

- سوال اصلی: آیا همیشه می توان خط مزبور را پیدا کرد؟
  - هر دو رابطه یاد شده وجود داشته باشد؟
- اگر خاصیت جدایی پذیری خطی وجود داشته باشد. یعنی اگر همه داده های یادگیری، در فضای مشخصه ها، توسط یک رابطه خطی از هم قابل تفکیک باشند، مسئله دارای بیش از یک جواب است.
- در صورت وجود جدایی پذیری خطی، بر حسب میزان فاصله نزدیکترین نمونه ها به خط مزبور باید یک مقدار ترشهلد b انتخاب نمود تا نامعادله مزبور به یک معادله تبدیل شود.
  - حتی اگر جدایی پذیری خطی وجود نداشته باشد، می توان معیاری مانند کمترین میزان خطا را در نظر گرفت و بر اساس آن خط بهینه را پیدا کرد.

طبقه بندی روش های جستجوی خط بهینه

- روش های جستجوی جواب را می توان از دو منظر تقسیم بندی نمود.
- از منظر تاثیر نقش داده ها (در معیار جستجو):
  - روش های یکپارچه
  - روش های نمونه به نمونه
- از منظر روش جستجو:
  - روش های فرموله (تک گام)
  - روش های گام به گام

روش یکپارچه تک گام

One step Batch technique

روش شبه معکوس

$$A \times W = b$$

using the pseudo inverse technique

$$\mathrm{A}^{\mathrm{T}} \times A \times W = A^{T} \times b$$

assum the matrix is nonsingular :

$$(\mathrm{A}^{\mathrm{T}} \times \mathrm{A})^{-1} \times (\mathrm{A}^{\mathrm{T}} \times A) \times W = (\mathrm{A}^{\mathrm{T}} \times \mathrm{A})^{-1} \times A^{T} \times b$$

$$W = (\mathrm{A}^{\mathrm{T}} \times \mathrm{A})^{-1} \times A^{T} \times b$$

مشکلات عملی محتمل

- ممکن است ماتریس بدست آمده معکوس نداشته باشد.
- ممکن است ابعاد خیلی زیاد باشد.
- ممکن است برای تخمین اولیه ترشهلد b ایده خوبی نداشته باشیم
  - اگر b را خیلی کوچک بگیریم کار برای w با دشواری روبرو شویم
  - اگر b را خیلی بزرگ بگیریم برای یافتن w با دشواری روبرو شویم

روش های یکپارچه تکراری (جستجو)

- این روشها معمولا مبتنی برگرادیان محلی و یا جستجوی رندوم است
- در روش های مبتنی برگرادیان محلی، عموما یک تابع هزینه risk تعریف می شود که تابع **W** بوده و آنگاه مقداری از **W** پیدا می شود که تابع مزبور را می نیمم کند
- فاکتور های اساسی در پروسه جستجو:
  - The risk function **J** and its relationship to **W**
  - Learning **parameters** like learning rate
  - The **criteria for stopping** the procedure

$$W^{(n+1)} = W^{(n)} - \alpha \frac{\partial J(W)}{\partial (W)}\bigg|_{W=W^{(n)}}$$

Perceptron criteria :

$$J_p(W) = -\sum_{x_i \in \hat{\text{error}}} \hat{x}^T_i . W$$

$$\nabla_W J_p(W) = -\sum_{x_i \in \hat{\text{error}}} \hat{x}^T_i$$

$$W^{(n+1)} = W^{(n)} + \alpha \sum_{x_i \in \hat{\text{error}}} \hat{x}^T_i$$

روش های غیر یکپارچه(نمونه به نمونه)

- در این روش ها برای عوض شدن W در هر بار، تنها یک نمونه بررسی می شود.
- در مقایسه این دو روش با هم، روش های یکپارچه خصوصیات زیر را دارند:

- instability of patterns
- lower needed memory
- All patterns have contribution in updating in each iteration
- More stable procedure but the hazard of local minima
- Easier to implement in computer

شرایط توقف پروسه

- برای توقف پروسه معمولا به سه روش عمل می شود:
- To reach a specified error
  - (difference between W in nth and (n+1)th iterations)
- To reach a specified error rate
  - Rate of reduction in difference between W in two iteration
- To reach a specified number of iterations

الگوریتم نمونه برای طرح کلاسیفایر خطی

Choosing an initial value for " b"

$$b \xrightarrow{put} b^{(n)}$$

$$A^T A W^{(n)} = A^T b^n$$

$$b^{(n+1)} = b^{(n)} - \rho \nabla_b J_H (W^{(n)}, b^{(n)})$$

$$= b^{(n)} + \rho (AW^{(n)} - b^{(n)})$$

only the positive b are acceptable so :

$$b^{(n+1)} = b^{(n)} + \rho \mathrm{C}_i^{(n)}$$

$$\mathrm{C}_i^{(n)} = \begin{cases} (AW^{(n)} - b^n)_i & \text{if } (AW^{(n)} - b^n)_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Generalized Linear Discriminant Functions

- ممکن است توابعی که می توانند کلاس ها را جدا کنند همیشه شکل خطی نداشته باشند
- ممکن است گاهی مرز های تصمیم نیاز به سطوح پیچیده و غیر خطی داشته باشد
- در چنین شرایطی با توجه به خصوصیات مطلوب سیستم های خطی از یکسو و محدودیت های یاد شده از سوی دیگر تعمیم هایی (مانند آنچه در پائین آمده) صورت داد:

$$g(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_N f_N(x) + w_{N+1} \qquad (1)$$

where $f_i(x)$, $1 \le i \le N$ are scalar functions of the pattern x,

$x \in R^n$ (Euclidean Space)

تعبیر دیگر

- ابتدا یک نگاشت غیر خطی از X به فضایی از مشخصه ها با جدایی پذیری بیشتر
- سپس یافتن یک تابع جدا ساز خطی در فضای جدید
- یکی از مهمترین مفاهیم در این حوزه توابع اساسی Basis function و مشخصا توابع اساسی شعاعی هستند

جدایی پذیری خطی و غیر خطی

- سوال اساسی:
  - آیا می توان مسئله ای را که جدایی پذیری خطی ندارد، (و با ابزار های خطی قابل حل نیست) به فضایی نگاشت نمود که در آن جدایی پذیری خطی داشته باشد؟
- مثال مسئله XOR:

$xor(x1,x2)=(x_1 OR\ x_2)\ AND\ NOT\ (x_1\ AND\ x_2)$

- این مسئله در فضای ورودی های باینری فاقد جدائی پذیری خطی است

X1 / x2	0	1
0	0	1
1	1	0

ایده اساسی

- در مسئله XOR اعضای یک کلاس حول(لااقل یکی از) دو مرکز هستند
- اعضای کلاس دیگر حول هیچیک از دو مرکز نیستند
- پس اگر توابعی غیر خطی، از فاصله از این دو مرکزتعریف کنیم، که با افزایش فاصله بسرعت افت کنند،
  - برای اعضای کلاس دیگر همه این توابع بسیار نزدیک صفر خواهد بود پس ترکیب خطی این توابع نیز نزدیک صفر است
  - برای اعضای کلاس مورد نظر لااقل بعضی(یکی) از این توابع مقدار نسبتا بزرگی دارد پس ترکیب خطی آنها بزرگتر از صفر است

نگاشت به فضایی با پیچیدگی بالا

- فرض کنیم ورودی ها را به فضایی با متغیر های جدید به شرح زیر نگاشت دهیم:

$$\varphi_1(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{t}_1\|^2}, \qquad \mathbf{t}_1 = [1, 1]^T$$

$$\varphi_2(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{t}_2\|^2}, \qquad \mathbf{t}_2 = [0, 0]^T$$

- در این شرایط چهار نقطه مورد نظر ما به صورت زیر نگاشت داده می شوند:

ادامه

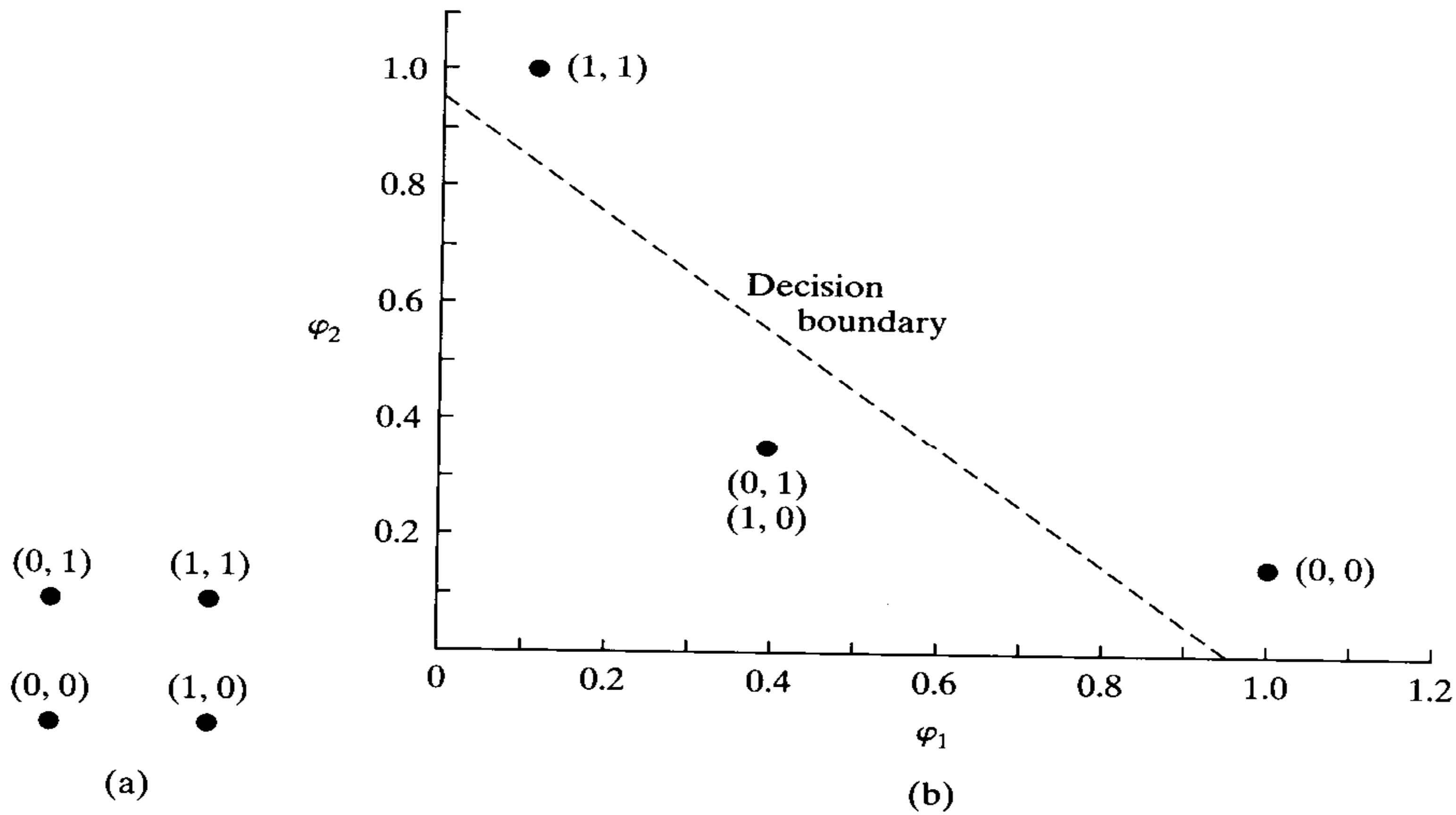

FIGURE 5.2 (a) The four patterns of the XOR problem; (b) Decision-making diagram.

ادامه

- با تعریف : $f_{n+1}(x) = 1$

$$g(x) = \sum_{i=1}^{N+1} w_i f_i(x) = w^T . \dot{x}$$

$$w = (w_1, w_2, \ldots, w_N, w_{N+1})^T$$

$$\dot{x} = (f_1(x), f_2(x), \ldots, f_N(x), f_{N+1}(x))^T$$

- در این نمایش، در واقع ما هر تابع $g(x)$ را(که دارای خصوصیات غیر خطی است) به فضایی با N+1 بعد برده ایم $(N+1>n)$.
- در *این فضا*، تلاش آن است که خط بخصوصی را در این فضا پیدا کنیم که جدایی سازی را انجام دهد.

معمولترین توابع :

- توابع چند حمله ای

$$g(x) = (\dot{w})^T \dot{x}$$

مثال :توابع درجه ۲

$$g(x) = w_1 x_1^2 + w_2 x_1 x_2 + w_3 x_2^2 + w_4 x_1 + w_5 x_2 + w_6$$

$$\mathrm{w} = (\mathrm{w}_1, w_2, \ldots, w_6)^T$$

$$\dot{\mathrm{x}} = (\mathrm{x}_1^2, x_1 x_2, x_2^2, x_1, x_2, 1)^T$$

ادامه

$$g(x) = \sum_{i=1}^{n} w_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j + \sum_{i=1}^{n} w_i x_i + w_{n+1} \qquad (2)$$

تعداد ترم های سمت راست عبارت است از:

$$l = N + 1 = n + \frac{n(n-1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}$$

د رواقع این تعداد وزن های مورد نیاز برای چنین نگاشتی است

- If for example n = 3, the vector $\dot{x}$ is 10-dimensional
- If for example n = 10, the vector $\dot{x}$ is 65-dimensional

در حالت وجود تابع جدا ساز چند جمله ای از درجه m $f_i(x)$

$$f_i(x) = x_{i_1}^{e_1} x_{i_2}^{e_2} \dots x_{i_m}^{e_m}$$
$$1 \le i_1, i_2, \dots, i_m \le n$$
$$1 \le i \le m \text{ is 0 or 1.}$$

- شکل کلی یک تابع چند جمله با درجه ای بین صفر تا m است و برای جلو گیری از تکرار معمولا قرار می دهیم:
- $$g^m(x) = \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \dots \sum_{i_m=i_{m-1}}^{n} w_{i_1 i_2 \dots i_m} x_{i_1} x_{i_2} \dots x_{i_m} + g^{m-1}(x)$$
- $g^0(x) = w_{n+1}$) معمولترین شکل است

Example 1: $n = 3$; $m = 2$:

$$g^2(x) = \sum_{i_1=1}^{3} \sum_{i_2=i_1}^{3} w_{i_1 i_2} x_{i_1} x_{i_2} + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4$$

$$= w_{11} x_1^2 + w_{12} x_1 x_2 + w_{13} x_1 x_3 + w_{22} x_2^2 + w_{23} x_2 x_3 + w_{33} x_3^2$$

$$+ w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4$$

Example 2: $n = 2$; $m = 3$:

$$g^3(x) = \sum_{i_1=1}^{2} \sum_{i_2=i_1}^{2} \sum_{i_3=i_2}^{2} w_{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} + g^2(x)$$

$$= w_{111} x_1^3 + w_{112} x_1^2 x_2 + w_{122} x_1 x_2^2 + w_{222} x_2^3 + g^2(x)$$

$$\textit{where } \; g^2(x) = \sum_{i_1=1}^{2} \sum_{i_2=i_1}^{2} w_{i_1 i_2} x_{i_1} x_{i_2} + g^1(x)$$

$$= w_{11} x_1^2 + w_{12} x_1 x_2 + w_{22} x_2^2 + w_1 x_1 + w_2 x_2 + w_3$$

شکل عمومی توابع درجه دوم، در نمایش برداری و ماتریسی

$$g(x) = x^T A x + x^T b + c$$

$$A = (a_{ij}), \qquad b = (b_1, b_2, \ldots, b_n)^T$$

c, depends on tw_{ii}, w_{ij}, w_i

- اگر A معین مثبت باشد، آنگاه تابع تصمیم (سطح جدا کننده) توسط بردار های ویژه A قابل تعیین است
- در شکل خاص که A ماتریس یکه است، تابع تصمیم یک صفحه خواهد بود.

خلاصه

- موضوع بحث این نشست، طراحی مستقیم کلاسیفایر از روی داده های یادگیری
- توابع جدا ساز خطی با استفاده مستقیم از داده ها
  - معرفی (ارائه لیست) مهمترین کلاسیفایر ها
  - طبقه بندی روش ها از روی معیار و روش جستجو
  - جدا یی پذیری خطی
- جدایی پذیری غیر خطی
- تعمیم روش های خطی به مسائلی که دارای جدایی پذیری خطی نیستند.