

Machine Learning

By: Dr. Omid Mahdi Ebadati

Ph.D. (Computer Science), Delhi



Kharazmi University

Kharazmi University

یادگیری

بر اساس نمونه

**Instance Based
Learning**

مقدمه IBL

○ روشهایی که تاکنون بررسی کردیم، سعی بر این بود که با استفاده از مثالهای آموزشی تابعی پیدا کنیم که بتواند توصیف کننده داده ها باشد.

○ در روش یادگیری IBL بسادگی فقط مثالها را ذخیره میکنیم و هرگونه تعمیم تا مشاهده مثال جدید به تعویق میافتد. به همین دلیل این روش گاهی روش تنبل یا lazy هم نامیده میشود.

مقدمه IBL

با مشاهده مثالهای جدید، رابطه آن با نمونه های ذخیره شده بررسی شده و یک مقدار برای تابع هدف آن نسبت داده میشود.

در روش IBL یک فرضیه عمومی مشخص برای داده ها بدست نخواهد آمد بلکه دسته بندی هر نمونه جدید هنگام مشاهده آن و بر اساس نزدیکترین مثالهای ذخیره شده، انجام خواهد شد.

IBL Introduction

- o In machine learning, **instance-based learning** or **memory-based learning** is a family of learning algorithms that, **instead of performing explicit generalization, compare new problem instances with instances seen in training, which have been stored in memory.**

IBL Introduction

- o It is called instance-based because it **constructs hypotheses directly from the training instances themselves.**
- o This means that the hypothesis **complexity can grow with the data:**
 - o in the **worst case**, a hypothesis is a list of n training items and the computational complexity of classification a single new instance is $O(n)$.

Instance-based Learning



Its very similar to a Desktop!!



Introduction (Cont.)

- o One **advantage** that instance-based learning has over other methods of machine learning is its **ability to adapt its model to previously unseen data.**
- o Where **other methods** generally **require** the **entire set of training data to be re-examined** when one instance is **changed**, **instance-based** learners may **simply store a new instance or throw an old instance away.**

Introduction (Cont.)

Local approximation to the target function that applies in **the neighborhood** of the query instance.

- o **Cost of classifying** new instances can be **high**: Nearly all **computations take place at classification time**
 - o Examples: *k*-Nearest Neighbors
- o **Radial Basis Functions: Bridge** between **instance-based learning** and **artificial neural networks**

IBL also Known as:

- o Also known as:
 - o Case-based
 - o Exemplar-based
 - o Nearest Neighbor
 - o Memory-based
 - o Lazy Learning

k-Nearest Neighbor

- o A simple example of an instance-based learning algorithm is the *k*-nearest neighbor algorithm.
- o **Daelemans** and Van den **Bosch** describe variations of this algorithm for use in **Natural Language Processing (NLP)**, claiming that memory-based learning is both more **psychologically realistic than other machine-learning schemes and more effective in practice.**

k-Nearest Neighbor Learning

k-Nearest Neighbor Learning

Instance $\mathbf{x} = [a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_n(\mathbf{x})] \in \mathcal{R}^n$

$d(\mathbf{x}_i, \mathbf{x}_j) = [(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)]^{1/2} = \text{Euclidean Distance}$

- o Discrete-Valued Target Functions

$$f : \mathcal{R}^n \rightarrow V = \{ v_1, v_2, \dots, v_s \}$$

تفاوت اساسی با مدل‌های دیگر

روش IBL برای هر نمونه جدید، تقریب جداگانه‌ای از تابع هدف را ایجاد می‌کند. این تقریب فقط به همسایگی نمونه جدید قابل اعمال بوده و هرگز نمیتواند بر روی فضای تمام نمونه‌ها عمل کند.

کاربرد این روش هنگامی موثر است که تابع هدف خیلی پیچیده بوده ولی در عین حال قابل نمایش توسط توابع ساده‌تر محلی باشد.

IBL When to Consider? Adv. vs Disadv.

When To Consider IBL

- o Instances map to points
- o Less than 20 attributes per instance
- o Lots of training data

Advantages:

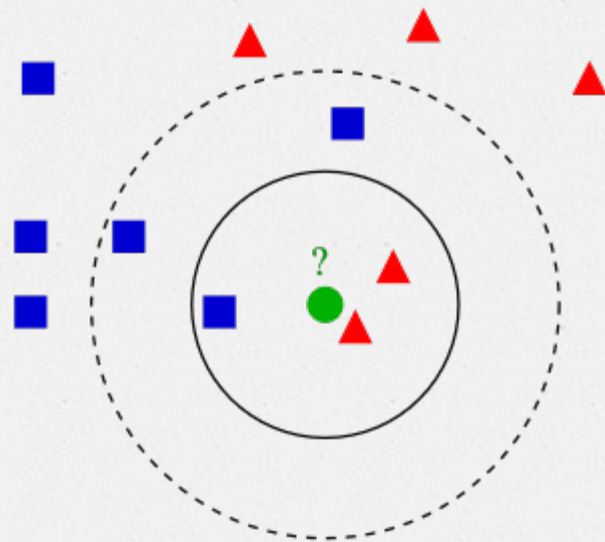
- o Training is very fast
- o Learn complex target functions
- o Don't lose information

Disadvantages:

- o Slow at query time
- o Easily fooled by irrelevant attributes

Ex.

The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).



مشخصه ها

این روش دارای 3 مشخصه اصلی است:

○ تابع شباهت :

مشخص میکند که دو نمونه چقدر نزدیک به هم هستند. انتخاب این تابع میتواند بسیار مشکل باشد. مثلا چگونه میتوان شباهت رنگ موی 2 نفر را بیان نمود؟

○ انتخاب نمونه ها برای ذخیره :

در این الگوریتم سعی میشود نمونه هائی ذخیره شوند که عمومی تر باشند. تشخیص اینکه آیا یک نمونه عمومیت دارد یا خیر، میتواند کار مشکلی باشد.

○ تابع دسته بندی کننده:

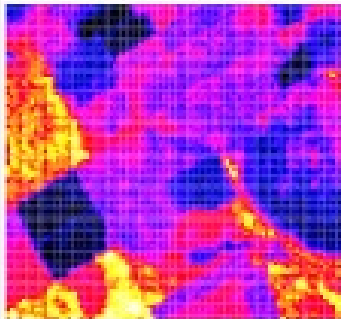
تابعی است که با مشاهده یک مثال دسته بندی آنرا تعیین میکند .

مشکلات

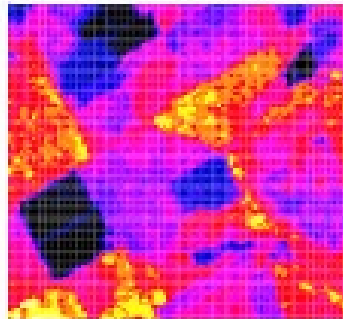
- دسته بندی داده جدید میتواند بسیار پرهزینه باشد. زیرا در مرحله آموزش عملی صورت نمی پذیرد و تمامی محاسبات در هنگام دسته بندی انجام میگردند.
- از اینرو برای کاهش زمان دسته بندی از تکنیک های ایندکس استفاده میشود.
- در اغلب روشهای IBL برای بازخوانی مثالهای مشابه از حافظه از تمامی ویژگی های موجود استفاده میشود. بنابراین اگر تابع هدف فقط به برخی از ویژگی ها بستگی داشته باشد، مثالهایی که واقعا مشابه هستند ممکن است بسیار از یکدیگر دور شوند.

Image Scene Classification

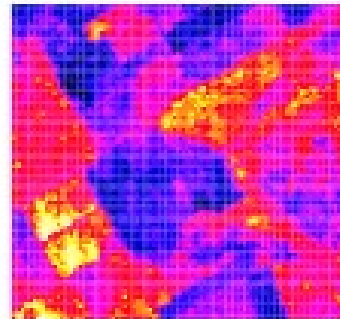
Spectral Band 1



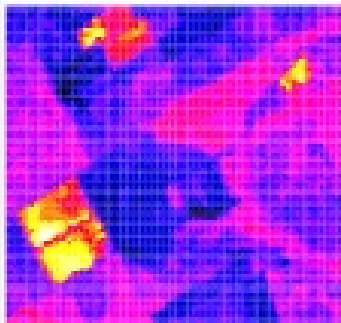
Spectral Band 2



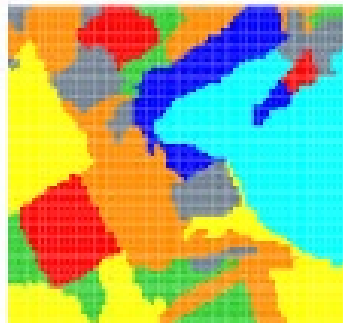
Spectral Band 3



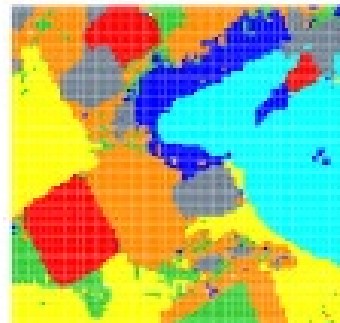
Spectral Band 4



Land Usage



Predicted Land Usage



برای هر تصویر با استفاده از مقادیر پیکسل‌های آن یک signature محاسبه شده و از آن برای مقایسه تصویر ورودی با تصاویر موجود در دیتا بیس استفاده میشود.

روشهای مختلف IBL

- 1. K-Nearest Neighbor (KNN)
 - Discrete Target Functions
 - Continuous Target Functions
 - Distance Weighted
- 2. Locally Weighted Regression
- 3. Radial Basis Function Networks
- 4. Case-Based Reasoning
- 5. General Regression Neural Networks

1. (k -NN)

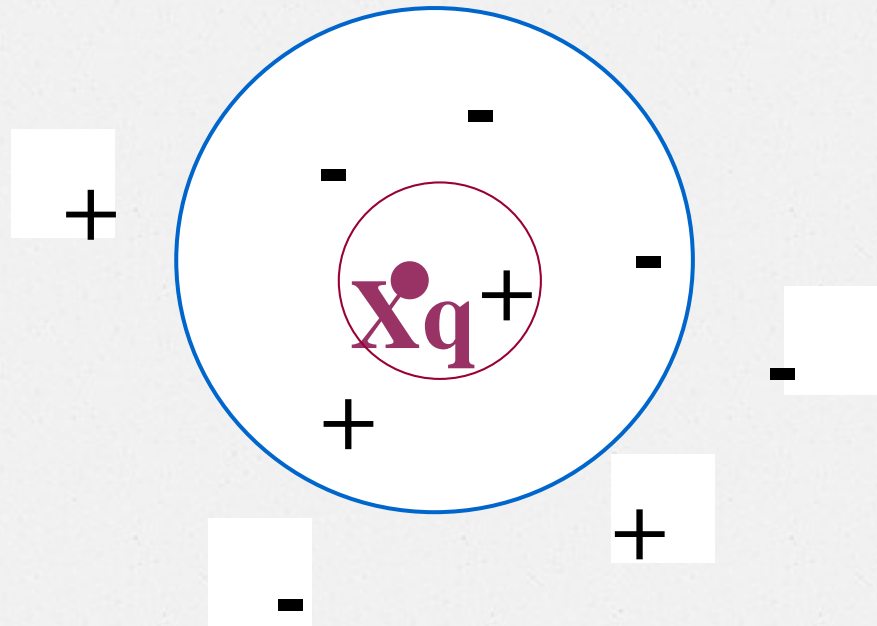
○ k -NN ساده ترین و متداولترین روش برپایه یادگیری نمونه است.

○ در این روش فرض میشود که تمام نمونه ها نقاطی در فضای n بعدی حقیقی هستند و همسایه ها بر مبنای فواصل اقلیدسی استاندارد تعیین میشوند.

○ k تعداد همسایه های در نظر گرفته شده است .

مثال

o اگر $k=1$ انتخاب شود، الگوریتم 1-NN مقدار نزدیکترین نمونه به X_q را انتخاب خواهد نمود. برای مقادیر بزرگتر k متداولترین مقدار بین k همسایه نزدیک انتخاب خواهد شد.



بایاس استقرا

o بایاس استقرا الگوریتم k -NN را میتوان بصورت زیر در نظر گرفت:

دسته بندی یک نمونه، مشابه دسته بندی نمونه های دیگری خواهد بود که در نزدیکی آن قرار دارند.

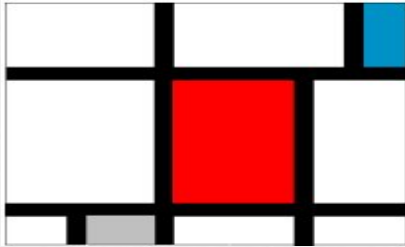
الگوریتم k -NN برای تابع هدف پیوسته

الگوریتم k -NN را میتوان بسادگی برای توابع هدف پیوسته نیز استفاده نمود. در این حالت بجای انتخاب متداولترین مقدار موجود در همسایگی، مقدار میانگین k مثال همسایه محاسبه میشود.

در نتیجه در خط آخر الگوریتم از رابطه زیر استفاده میشود:

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k f(x_i)}{k}$$

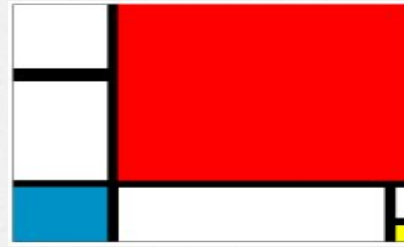
مثال



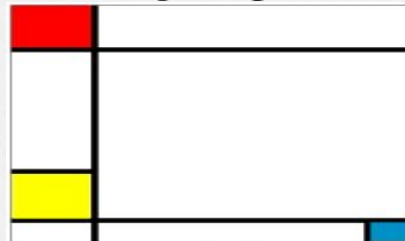
one



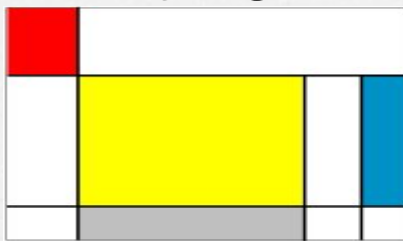
two



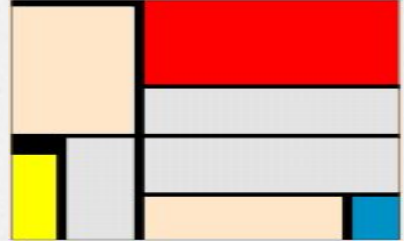
three



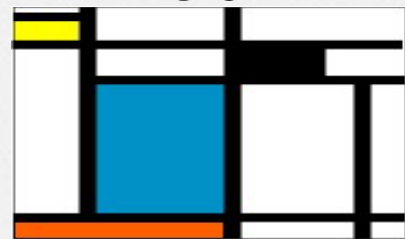
four



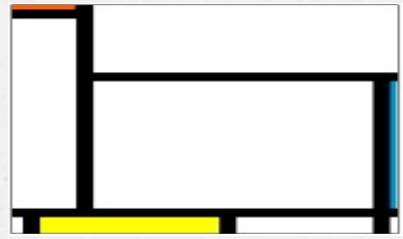
five



six



seven



Eight ?

برخی از این نقاشی ها طبق جدول بعد متعلق به نقاشی به نام Mondrian هستند. مشخص کنید که آیا نقاشی هشتم نیز به وی تعلق دارد؟

Training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	6	1	10	4	No
2	4	2	8	5	No
3	5	2	7	4	Yes
4	5	1	8	4	Yes
5	5	1	10	5	No
6	6	1	8	6	Yes
7	7	1	14	5	No

Test instance

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	7	2	9	4	

نرمالیزه کردن داده های آموزشی

یک راه نرمالیزه کردن داده آموزشی $a_r(x)$ به $a'_r(x)$ عبارت است از

$$x'_t \equiv \frac{x_t - \bar{x}_t}{\sigma_t}$$

$\bar{x}_t \equiv \text{mean of } t^{\text{th}} \text{ attributes}$

$\sigma_t \equiv \text{standard deviation of } t^{\text{th}} \text{ attributes}$

Normalised training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	0.632	-0.632	0.327	-1.021	No
2	-1.581	1.581	-0.588	0.408	No
3	-0.474	1.581	-1.046	-1.021	Yes
4	-0.474	-0.632	-0.588	-1.021	Yes
5	-0.474	-0.632	0.327	0.408	No
6	0.632	-0.632	-0.588	1.837	Yes
7	1.739	-0.632	2.157	0.408	No

Test instance

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	1.739	1.581	-0.131	-1.021	

Distance-weighted k-NN

میتوان عملکرد این الگوریتم را با در نظر گرفتن وزنی برای هر یک از k مثال همسایگی بهتر نمود. این وزن بر اساس فاصله نمونه ها تا نمونه مورد بررسی اعمال میشود و معمولاً با فاصله نمونه ها رابطه معکوس دارد.

$$\hat{f}(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k w_i \delta(v, f(x_i)) \quad \text{where } w_i = \frac{1}{d(x_q, x_i)^2} \quad \text{در حالت گسسته: } \circ$$

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i} \quad \text{where } w_i = \frac{1}{d(x_q, x_i)^2} \quad \text{در حالت پیوسته } \circ$$

در صورت اعمال وزن این امکان وجود خواهد داشت که به جای k نمونه همسایه، از تمامی نمونه ها برای دسته بندی استفاده کنیم. اما این انتخاب باعث کند شدن عمل دسته بندی خواهد شد.

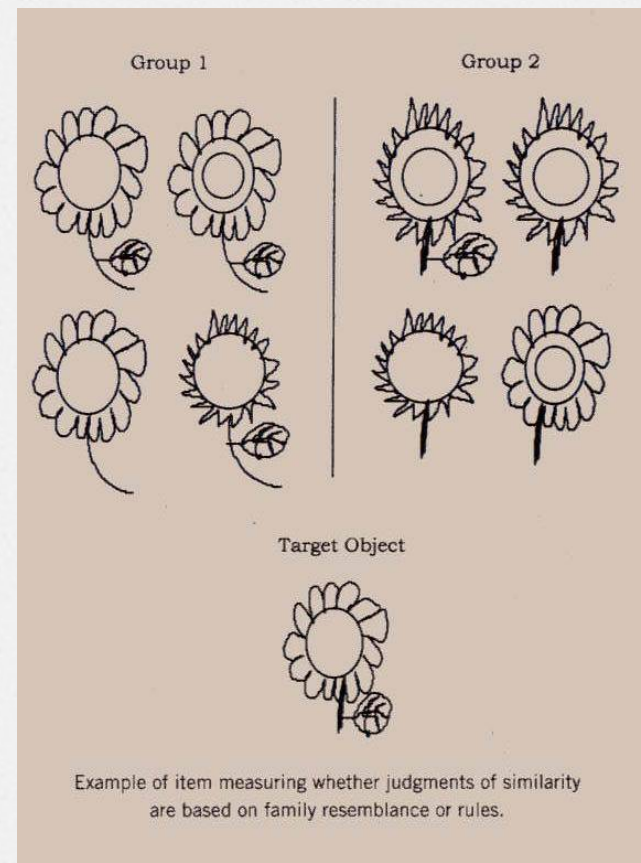
مزیت DW- k-NN

○ الگوریتم Distance-weighted k-NN بطور موثری در مسائل عملی مختلفی برای استنتاج استقرائی بکار رفته است.

○ این روش نسبت به نویز مقاوم بوده و در مواردی که داده آموزشی زیادی موجود باشد بسیار کارآمد است.

Rules and Instances in Human Learning Biases

- Psychological experiments show that people from different cultures exhibit distinct categorization biases.
- “Western” subjects favor simple rules (straight stem) and classify the target object in group 2.
- “Asian” subjects favor global similarity and classify the target object in group 1.



The curse of dimensionality

از آنجائیکه برای محاسبه فاصله از تمامی ویژگی ها استفاده میشود این امکان وجود دارد که حتی ویژگی های نامرتب در امر دسته بندی مورد استفاده قرار گیرند. این امر برخلاف روشهایی مثل درخت تصمیم است که در آن سعی میشود تا فقط از ویژگی های مرتبط استفاده شود.

برای مثال فرض کنید که هر نمونه با 20 ویژگی مشخص شوند که از میان آنان فقط 2 ویژگی برای دسته بندی کافی باشند در اینصورت ممکن است نمونه های ذخیره شده ای که در ایندو ویژگی مشابه هستند بسیار از هم فاصله داشته باشند. در اینصورت معیار فاصله مورد استفاده در k-NN میتواند بسیار گمراه کننده باشد.

این مسئله curse of dimensionality نامیده میشود.

Cross-validation

○ یک راه حل این مشکل استفاده از وزن بیشتر برای ویژگی های مرتبط است. این امر مشابه تغییر مقیاس محور هاست: محور ویژگی های مرتبط کوتاهتر و محور ویژگی های نامرتب طولانی تر میشوند.

○ برای تعیین وزن ویژگی ها میتوان از روش **cross-validation** استفاده نمود:

○ مجموعه ای از داده ها به عنوان داده های آموزشی انتخاب میشوند.

○ مقادیر z_1, \dots, z_n بعنوان ضرایبی که باید در هر محور ضرب شوند انتخاب میگردند. این انتخاب بنحوی است که خطای دسته بندی در باقیمانده مثالها کاهش یابد.

○ میتوان با قرار دادن $z_j=0$ اثر یک ویژگی را بکلی حذف نمود.

Indexing & kd-tree way

○ از آنجائیکه در روش k -NN دسته بندی مثالها تا زمان برخورد با آن مثال به تعویق میافتد استفاده از **Indexing** برای مرتب کردن مثالهای آموزشی میتواند بطور چشمگیری کارایی الگوریتم را افزایش دهد.

○ روش kd -tree یک روش برای ایندکس کردن است که در آن نمونه ها در سطح یک درخت ذخیره شده و نمونه های نزدیک به هم در همان گره و یا گره های نزدیک به هم ذخیره میشوند.

2. Locally Weighted Regression

○ الگوریتم LWR تعمیمی بر الگوریتم K-NN است که تقریب صریحی از تابع f حول ناحیه محلی در برگیرنده نمونه مورد بررسی X_q بدست میدهد.

○ این تقریب محلی با استفاده از مثالهای نزدیک هم و یا مثالهای distance-weighted انجام میشود.

○ این تابع ممکن است یک تابع خطی، درجه دو و یا یک شبکه عصبی باشد.

دلیل نامگذاری:

○ Local: از مثالهای نزدیک نمونه مورد بررسی استفاده میکند

○ Weighted: اثر هر مثال آموزشی با در نظر گرفتن فاصله آن منظور میشود

○ Regression: برای تقریب یک تابع با مقدار حقیقی بکار میرود

Locally Weighted Regression

این روش از یک تابع خطی برای تقریب تابع هدف در نزدیکی مثال مورد بررسی استفاده میکند:

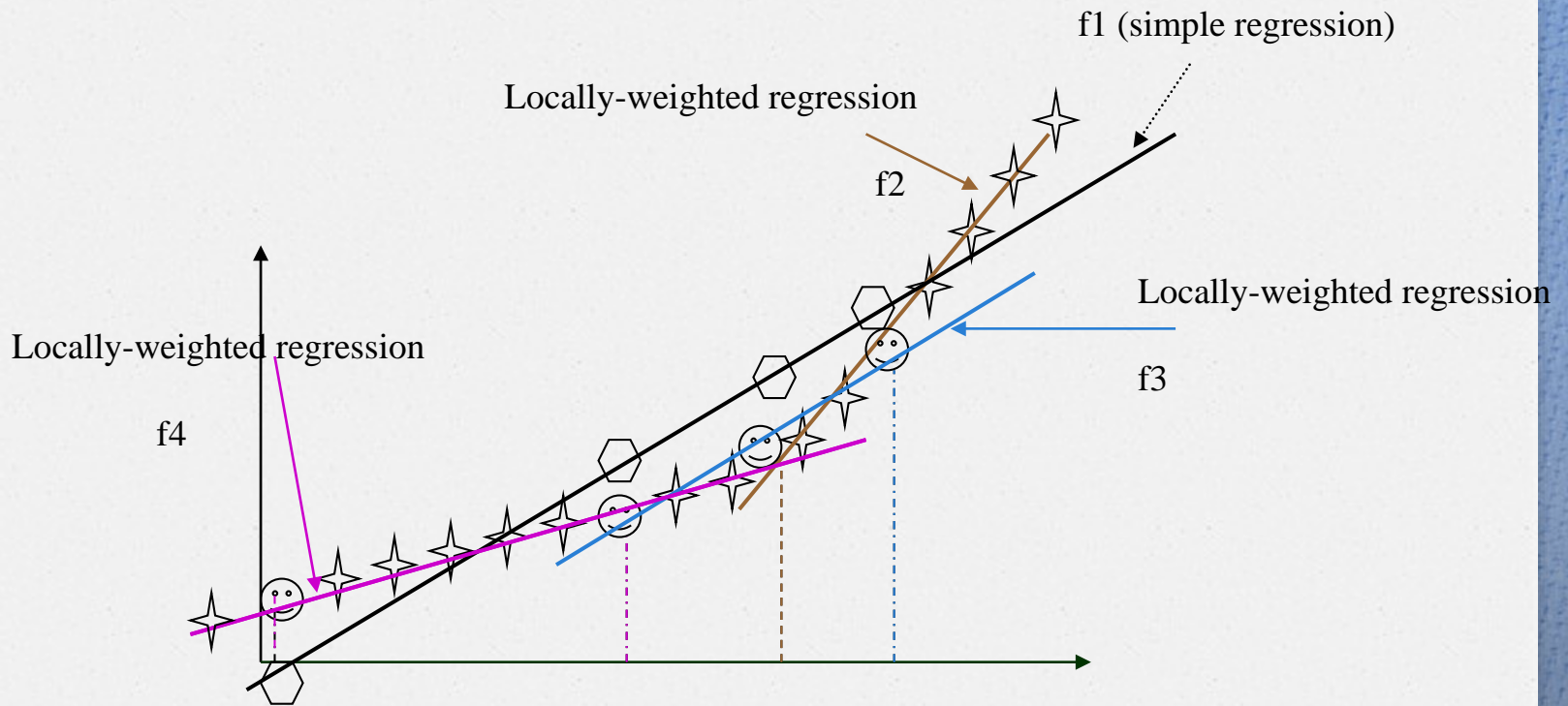
$$\hat{f}(x) = w_0 + w_1 a_1(x) + \dots + w_n a_n(x)$$

این تابع مشابه تابع مورد استفاده در محاسبه وزنهای یک شبکه عصبی است که در آن وزنها طوری انتخاب میشوند که مقدار خطای زیر حداقل گردد:

$$E(\vec{w}) = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

که برای اینکار از قانون آموزش gradient descent استفاده میشود.

$$\Delta w_i = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_i(x)$$



Training data



Predicted value using simple regression



Predicted value using locally weighted (piece-wise) regression

انتخاب مقدار k

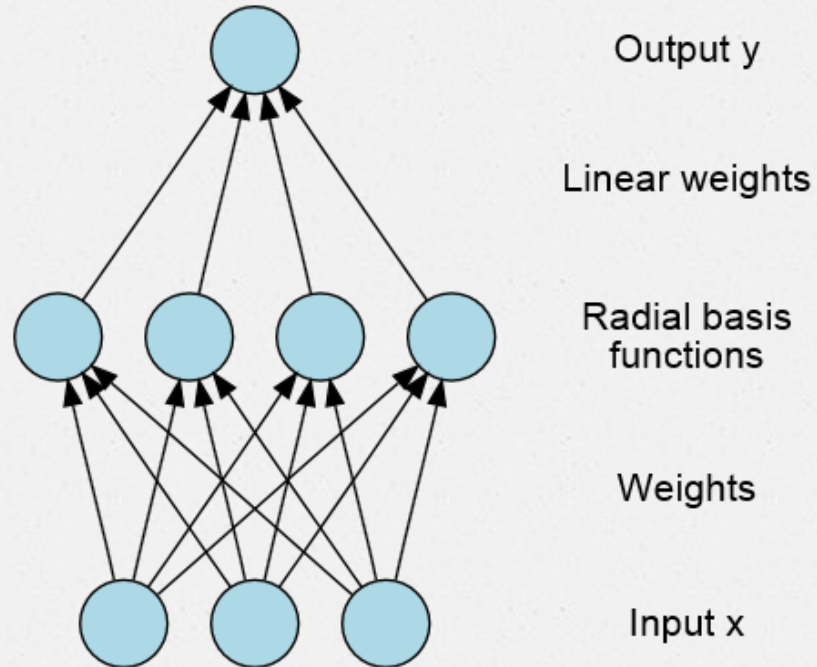
○ اگر k خیلی کوچک باشد، نسبت به نویز حساس خواهد بود
○ اگر K خیلی بزرگ باشد ممکن است یک همسایگی نقطه‌ای از سایر کلاسها را نیز در بر بگیرد.

- Large k :
 - less sensitive to noise (particularly class noise)
 - better probability estimates for discrete classes
 - larger training sets allow larger values of k
- Small k :
 - captures fine structure of space better
 - may be necessary with small training sets
- **Balance** must be struck **between large and small k**
- As training set approaches infinity, and k grows large, k NN becomes Bayes optimal

3. Radial Basis Function network

- o A **radial basis function network** is an **artificial neural network** that uses radial basis functions as activation functions.
- o **The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.** Radial basis function networks have many **uses**, including **function approximation**, **time series prediction**, classification, and system control.

RBF (Ex.)



4. Case-Based Reasoning (CBR)

- CBR is **similar to k-NN** methods in that:
 - They are *lazy* learning **methods** in that they defer **generalization** until a query comes around.
 - They **classify new query instances by analyzing similar instances while ignoring instances that are very different from the query.**
- However, CBR is different from k-NN methods in that:
 - They do **not** represent instances as **real-valued points**, but instead, they use a *rich symbolic* representation.
- **CBR** can thus be applied to **complex conceptual problems** such as the **design of mechanical devices or legal reasoning.**

Remarks on Lazy and Eager Learning

Lazy Learning: stores data and postpones decisions until a new query is presented:

- o k-Nearest Neighbour, Case based reasoning

Eager Learning: generalizes beyond the training data before a new query is presented:

- o Radial Basis Function Networks, ID3, ...
- o Lazy methods may consider the query instance \mathbf{x} when deciding how to generalize beyond the training data D (local approximation)
- o Eager methods cannot (they have already chosen their global approximation to the target function)

ویژگیهای یادگیری نمونه

○ مزایا:

- میتواند توابع پیچیده را مدل کند
- اطلاعات موجود در مثالهای آموزشی از بین نمیرود
- میتواند از نمایش سمبلیک نمونه ها استفاده کند

○ معایب:

- بازده الگوریتم هنگام انجام دسته بندی کم است
- تعیین یک تابع فاصله مناسب مشکل است
- ویژگیهای نامرتبب تاثیر منفی در معیار فاصله دارند
- ممکن است به حافظه بسیار زیادی نیاز داشته باشد

Conclusions

- o IBL methods **classify** test instances **based on similarity to specific training instances rather than forming explicit generalizations.**
- o **Typically trade decreased training time for increased testing time.**

Conclusions

- o *Instance Based Learning (IBL) results in classifying a new instance by examining and comparing it to the rest of the instances in the dataset.*
- o *An example of this type of learning is the **K-Nearest Neighbor** algorithm which is based on examining an average Euclidian distance of the nearest k neighbors' parameters given a certain situation.*

موفق باشید