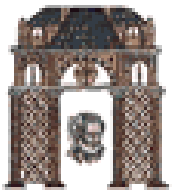
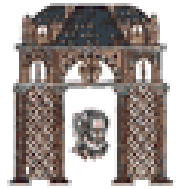


Αναγνώριση Προτύπων

Μη παραμετρικές τεχνικές
(Non Parametric Techniques)

Καθηγητής Χριστόδουλος Χαμζάς

Τα περιεχόμενα των παρουσιάσεων προέρχονται από τις παρουσιάσεις του αντίστοιχου διδακτέου μαθήματος του καθ. Παναγιώτη Τσακαλίδη, Τμ. Επιστήμης Υπολογιστών, Παν. Κρήτης. Το πρωτογενές υλικό βρίσκεται στην σελίδα http://www.csd.uoc.gr/~hy473/ *και βασίζεται στο βιβλίο: "Pattern Classification", R.O. Duda, P.E. Hart, D.G. Stork, Wiley, 2nd Ed., 2001*



Μη Παραμετρικές Τεχνικές

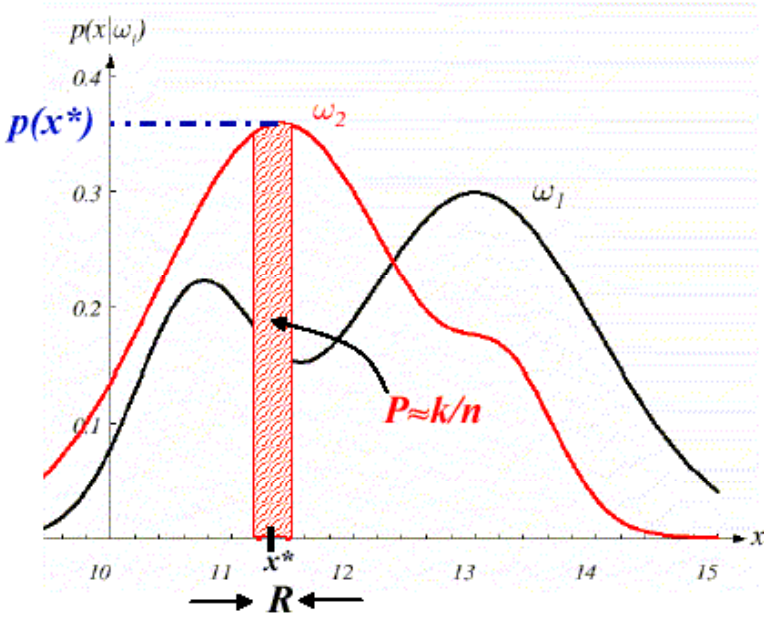
- Προβλήματα παραμετρικών τεχνικών:
  - Συνήθως δεν είναι γνωστή η μορφή της κατανομής.
  - Στην πράξη οι περισσότερες κατανομές είναι multimodal (περισσότερα από ένα μέγιστα), ενώ τα μοντέλα που χρησιμοποιούνται είναι unimodal.
  - Η προσέγγιση των πολυδιάστατων κατανομών σαν γινόμενο μονοδιάστατων δεν δουλεύει τόσο καλά στην πράξη.
- Μη παραμετρικές τεχνικές: Εκτίμηση της συνάρτησης κατανομής από το μηδέν.
  - Εκτίμηση των δεσμευμένων σ.π.π. $p(x/\omega_i)$ από τα δεδομένα μέσω γενίκευσης του *πολυδιάστατου ιστογράμματος*.
  - Απευθείας εκτίμηση των εκ των υστέρων πιθανοτήτων $P(\omega_i/x)$ και των συναρτήσεων διάκρισης.

Εκτίμηση Κατανομών

- Βασίζονται στο ότι η πιθανότητα ένα δείγμα $\boldsymbol{x}$ να βρίσκεται εντός της περιοχής R δίνεται από τη σχέση

$$P = P(\mathbf{x} \in R) = \int_{\mathbf{x}' \in R} p(\mathbf{x}')d\mathbf{x}'$$

- Αυτό το ολοκλήρωμα μπορεί να προσεγγισθεί είτε από το γινόμενο της τιμής $p(\mathbf{x})$ με το εμβαδόν της περιοχής, είτε από το πλήθος των δειγμάτων που βρίσκονται εντός της περιοχής



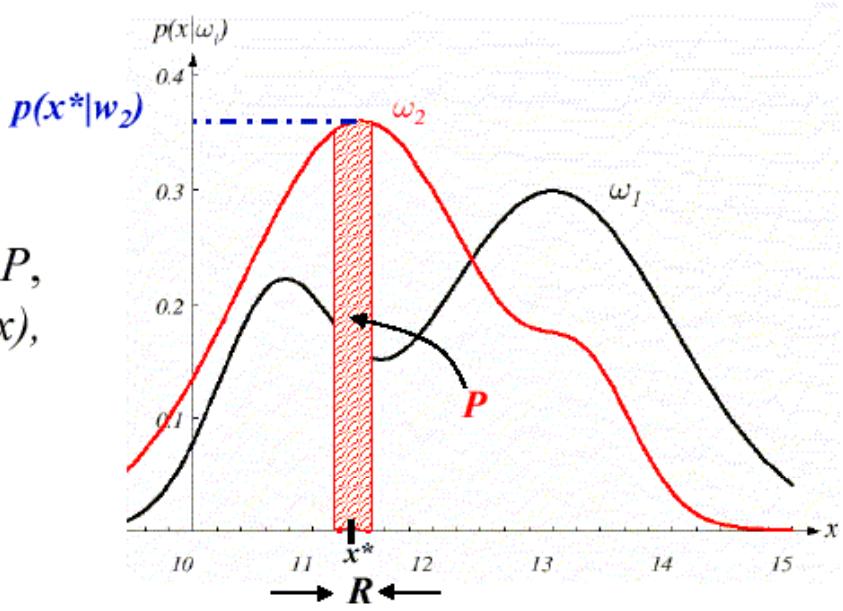
$$P = \int_{\mathbf{x}' \in R} p(\mathbf{x}')d\mathbf{x}' \approx p(\mathbf{x}^*) \cdot V \approx k/n$$

V: Εμβαδό της περιοχής R. Στην μονοδιάστατη περίπτωση, V= μήκος του R
k: Πλήθος δειγμάτων που βρίσκονται εντός της περιοχής R
n: Συνολικό πλήθος δειγμάτων

Εκτίμηση Κατανομών

Για την εκτίμηση της κατανομής στο $\boldsymbol{x}$, επιλέγουμε μια σειρά περιοχών $R_1, R_2, \ldots, R_n$ που περιέχουν το $\boldsymbol{x}$, όπου η R_i χρησιμοποιείται για i δείγματα. Έστω V_n ο όγκος της R_n, k_n το πλήθος των δειγμάτων πού βρίσκονται εντός της n-στής περιοχής, και $p_n(\boldsymbol{x})$ η n-στή εκτίμηση της $p(\boldsymbol{x})$. Τότε,

$$p_n(\mathbf{x}) = \frac{k_n / n}{V_n} \xrightarrow[n\to\infty]{} p(\mathbf{x})$$

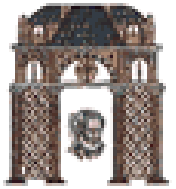


Για να είναι το k/n μια καλή εκτίμηση του P, και άρα το $p_n(x)$ μια καλή εκτίμηση του $p(x)$, θα πρέπει να ικανοποιούνται τα εξής:

$$\lim_{n\to\infty} V_n = 0$$

$$\lim_{n\to\infty} k_n = \infty$$

$$\lim_{n\to\infty} k_n / n = 0$$



Εκτίμηση Κατανομών

- Υπάρχουν δύο τρόποι για τη δημιουργία ακολουθιών περιοχών R_i ώστε να συγκλίνει η $p_n(\mathbf{x})$ στην $p(\mathbf{x})$:

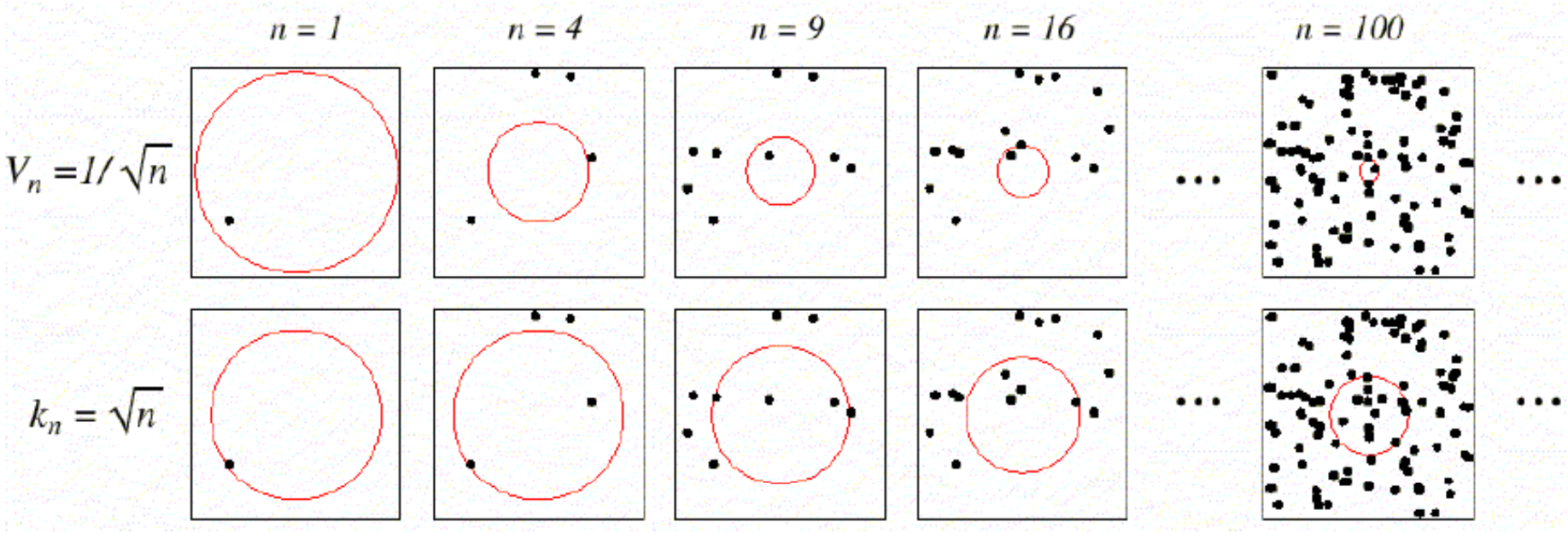
  - Μειώνουμε τον όγκο μιας αρχικής περιοχής ορίζοντας μια ακολουθία όγκων V_n ως συναρτήσεων του n, π.χ. $V_n = V_1/\sqrt{n}$

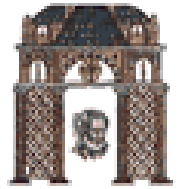
    → Εκτίμηση πυκνότητας με τη μέθοδο των Παραθύρων Parzen (Parzen Windows)

  - Ορίζουμε το k_n σαν συνάρτηση του n, $k_n = \sqrt{n}$ οπότε το V_n αυξάνει έως ότου περιλάβει k_n δείγματα.

    → Εκτίμηση πυκνότητας με τη μέθοδο των k_n Πλεισιέστερων Γειτόνων (k_n-Nearest Neighbor)

Δύο προσεγγίσεις





Παράθυρα Parzen

Συνάρτηση Παραθύρου Υπερκύβου

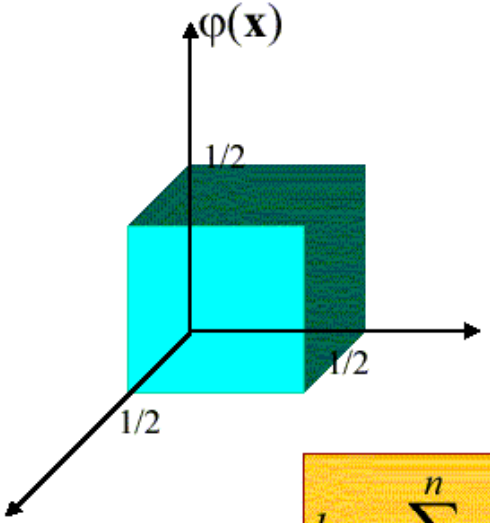
$$V_n = h_n^d$$

$$\varphi(u) = \begin{cases} 1 & |u_j| \le \frac{1}{2} \ \text{για} \ j = 1,2,\cdots,d \\ 0 & \alpha\lambda\lambda o\iota\acute{\omega}\varsigma \end{cases}$$

Παράθυρα Parzen

- Βασίζεται στην απαρίθμηση του πλήθους δειγμάτων που βρίσκονται μέσα σε μια δεδομένη περιοχή, με την περιοχή να μικραίνει καθώς το πλήθος δειγμάτων αυξάνει.

- Το πλήθος των δειγμάτων εντός της περιοχής υπολογίζεται με τη βοήθεια μιας συνάρτησης παραθύρου, του παραθύρου Parzen.



$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

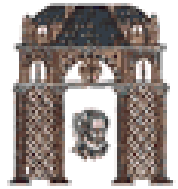
Ο όγκος της συνάρτησης, $V_n=(h_n)^d$

Συνάρτηση παραθύρου

Το πλάτος της συνάρτησης

$$k_n = \sum_{i=1}^{n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

Πλήθος δειγμάτων εντός της R_n, όπου η R_n έχει κέντρο **x** και πλάτος h_n



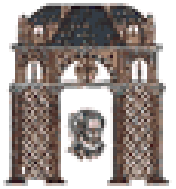
Παράθυρα Parzen

$$p_n(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)=\frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i) \quad \text{όπου} \quad \delta_n(\mathbf{x})=\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

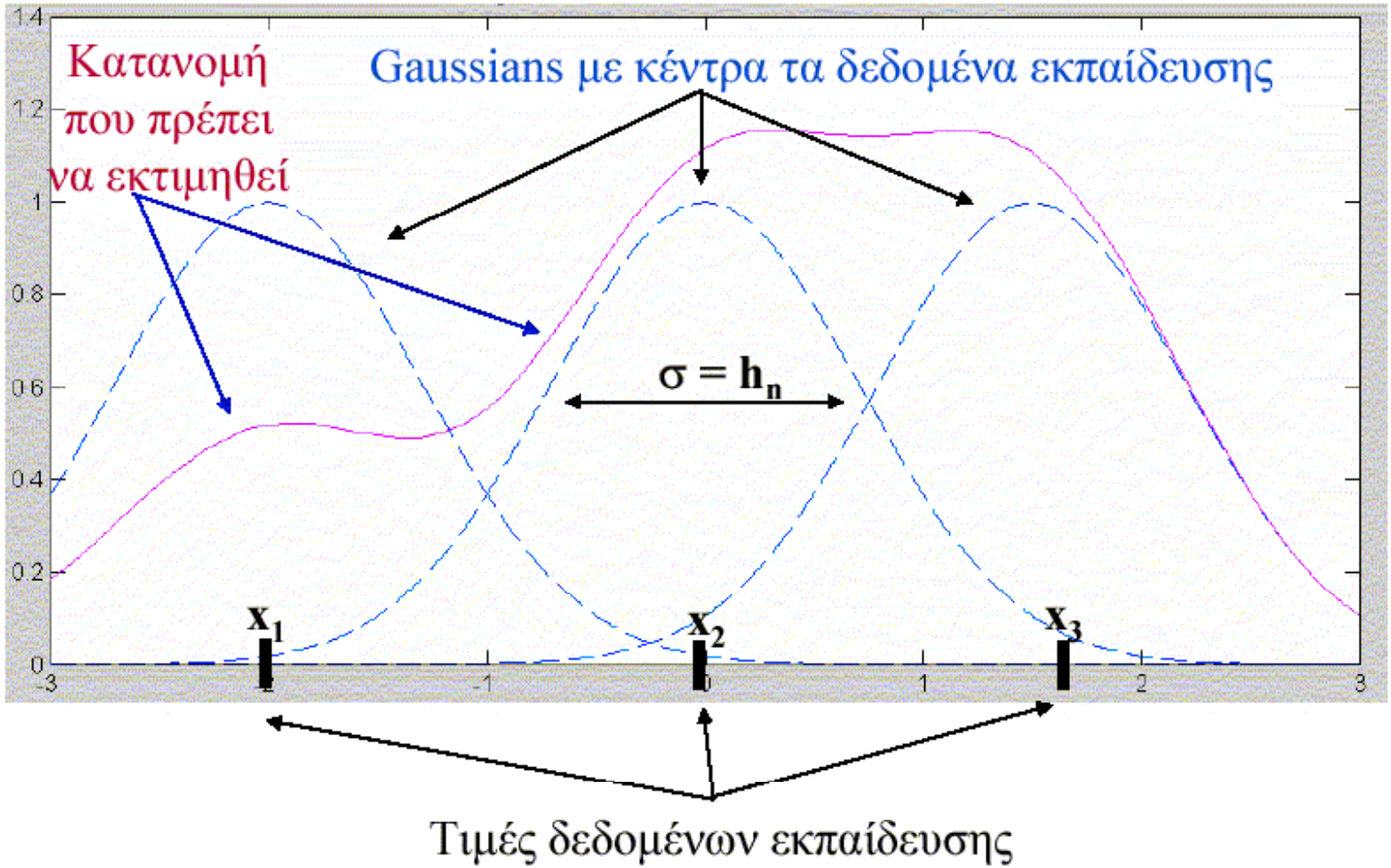
- Το παράθυρο $\varphi(.)$ μπορεί να είναι μια γενική συνάρτηση, όχι απαραίτητα υπερκύβος. Για να είναι η $p_n(x)$ μια έγκυρη σ.π.π. για κάθε n, θα πρέπει

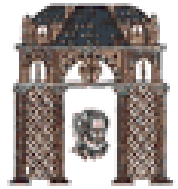
$$\varphi(\mathbf{u})\geq 0 \quad \text{και} \quad \int_{\mathbf{u}}\varphi(\mathbf{u})d\mathbf{u}=1$$

- Η $p_n(x)$ είναι ένας γραμμικός συνδυασμός των $\varphi(.)$, όπου κάθε δείγμα $\boldsymbol{x}_i$ συμβάλει στην εκτίμηση της $p(\boldsymbol{x})$ σύμφωνα με την απόστασή του από το $\boldsymbol{x}$. Εάν η $\varphi(.)$ είναι η ίδια μια έγκυρη σ.π.π., τότε η $p_n(x)$ θα συγκλίνει στην $p(x)$ καθώς το n αυξάνει. Μια τυπική επιλογή της $\varphi(.)$ είναι –σωστά μαντέψατε– η Γκαουσσιανή!
- Η $p(x)$ υπολογίζεται απλά σαν μία υπέρθεση Γκαουσσιανών, όπου κάθε Γκαουσσιανή είναι επικεντρωμένη στο αντίστοιχο δείγμα εκπαίδευσης. Η παράμετρος h_n είναι η διακύμανση της Γκαουσσιανής!!!



Παράθυρα Parzen





Επίδραση του πλάτους του παραθύρου, h_n

Για να υπολογίσουμε την pdf pn(x) στο σημείο **x** προσθέτουμε το

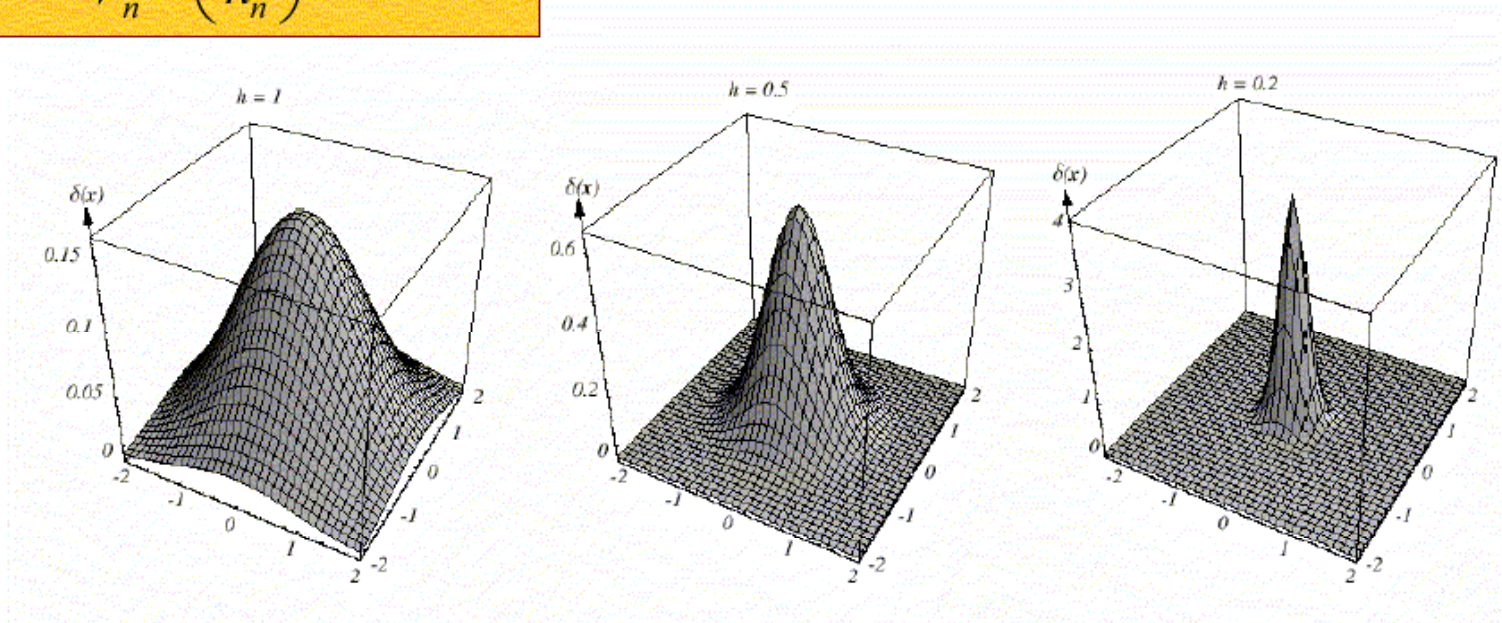
$$\varphi\left(\frac{\mathbf{x}-\mathbf{x_n}}{h_n}\right)$$

για όλα τα N σημεία $\mathbf{x_n}$ και το κανονικοποιούμε διαιρώντας με το $N V_N = N(h_N)^d$ δηλαδή

$$p_N(x) = \frac{1}{NV_N}\sum_{i=1}^{N}\varphi\left(\frac{\mathbf{x}-\mathbf{x_i}}{h_N}\right)$$

Επίδραση του πλάτους του παραθύρου, h_n

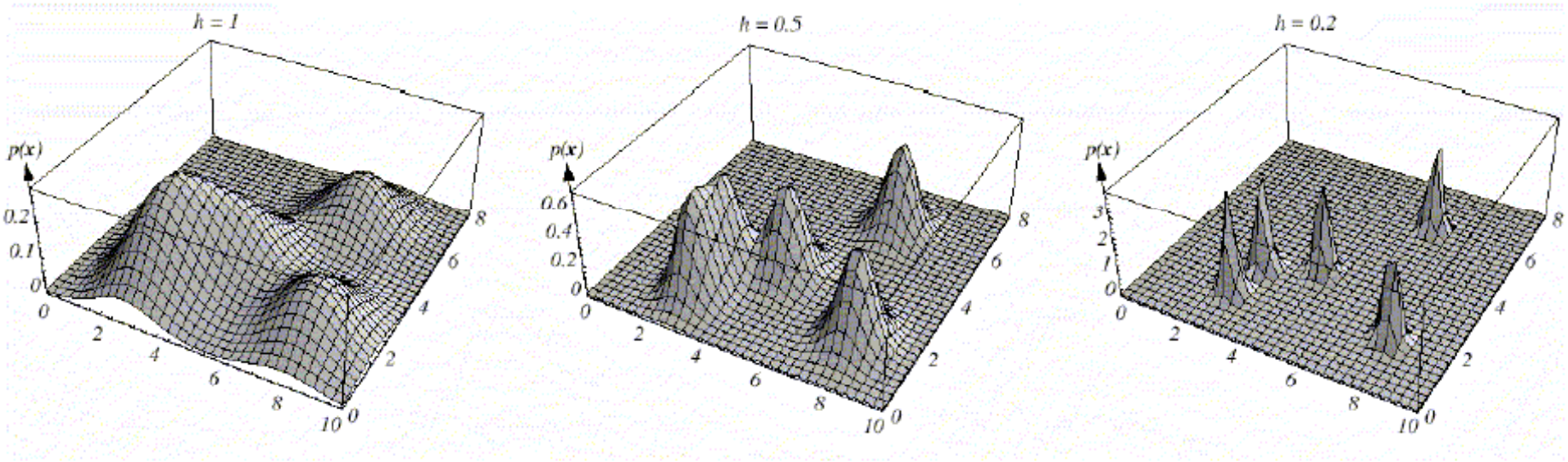
$$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right), \quad V_n = h_n^d$$



Η παράμετρος h_n επιδρά και στο πλάτος του παραθύρου αλλά και στο μέτρο του:

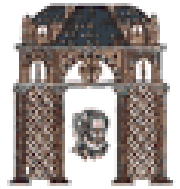
- **Όταν το h_n είναι μεγάλο (μικρό), το παράθυρο είναι πλατύ (στενό), το μέτρο του παραθύρου είναι μικρό (μεγάλο) και το x πρέπει να είναι μακρυά (κοντά) από το x_i πρίν η τιμή της συνάρτησης $\delta_n(x-x_i)$ αλλάξει αρκετά από την μέγιστη τιμή της $\delta_n(0)$.**

Επίδραση του πλάτους του παραθύρου, h_n



Πώς το πλάτος του παραθύρου επιδρά στην εκτίμηση της σ.π.π. *p(x)* :

- ✓ **Όταν το h_n είναι μεγάλο, η εκτιμήτρια $p_n(x)$ είναι η υπέρθεση *n* πλατιών συναρτήσεων επικεντρωμένων στα δείγματα εκπαίδευσης και αποτελεί μία ομαλή, "out-of-focus" εκτίμηση του *p(x)*, χωρίς μεγάλη ανάλυση.**
- ✓ **Όταν το h_n είναι μικρό, η $p_n(x)$ είναι η υπέρθεση *n* στενών συναρτήσεων, μία θορυβώδης, "erratic or noisy" εκτίμηση του *p(x)*.**



Ιδιότητες σύγκλισης της $p_n(\mathbf{x})$

- Μέση τιμή της τ.μ. $p_n(x)$:

$$\bar{p}_n(\mathbf{x}) = E[p_n(\mathbf{x})] = \int \delta_n(\mathbf{x}-\mathbf{v}) p(\mathbf{v}) d\mathbf{v}$$

είναι η συνέλιξη της $p(x)$ με τη συνάρτηση παραθύρου, δηλαδή είναι μία «θαμπή» (blurred) παραλλαγή της $p(x)$.

- Ισχύει ότι,

$$\delta_n(\mathbf{x}-\mathbf{v}) \xrightarrow[V_n \to 0]{} \delta(\mathbf{x}-\mathbf{v}) \quad \text{οπότε} \quad \bar{p}_n(\mathbf{x}) \xrightarrow[n \to \infty]{} p(\mathbf{x})$$

- Διασπορά της τ.μ. $p_n(x)$:

$$\mathrm{var}(p_n(\mathbf{x})) = \frac{1}{nV_n} \int \frac{1}{V_n} \varphi^2\left(\frac{\mathbf{x}-\mathbf{v}}{h_n}\right) p(\mathbf{v}) d\mathbf{v} - \frac{1}{n} \bar{p}_n^2(\mathbf{x}) \le \frac{\sup(\varphi(\cdot)) \bar{p}_n(\mathbf{x})}{nV_n}$$

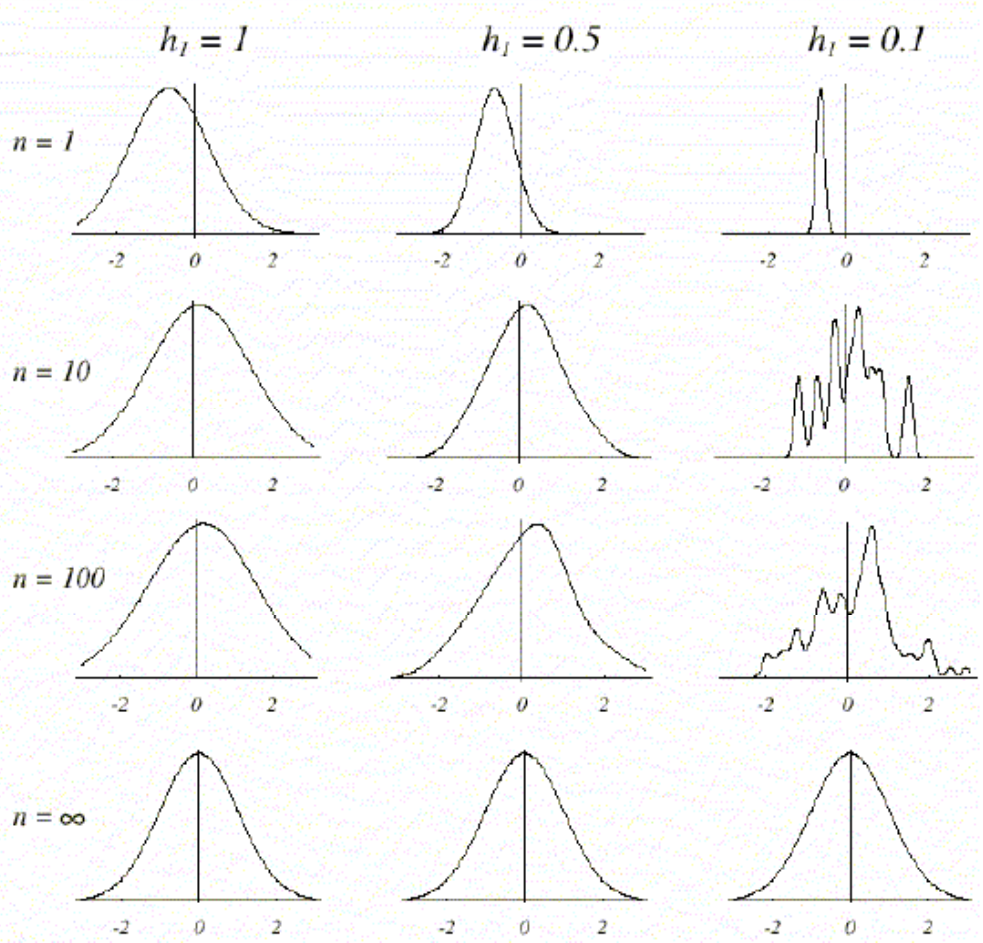
- Επομένως παίρνουμε μικρή διασπορά για μεγάλα V_n! Αλλά στο όριο, καθώς το $n \to inf$, μπορούμε να μειώσουμε το V_n πρός το 0 και η διασπορά να πηγαίνει και αυτή στο 0, αρκεί να ισχύει ότι $nV_n \to inf$.
- Δυνατές επιλογές: $V_n = V_1/\sqrt{n}$ ή $V_n = V_1/\ln n$

Παράδειγμα παραθύρων Parzen

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n}\varphi\left(\frac{x - x_i}{h_n}\right);$$

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$$

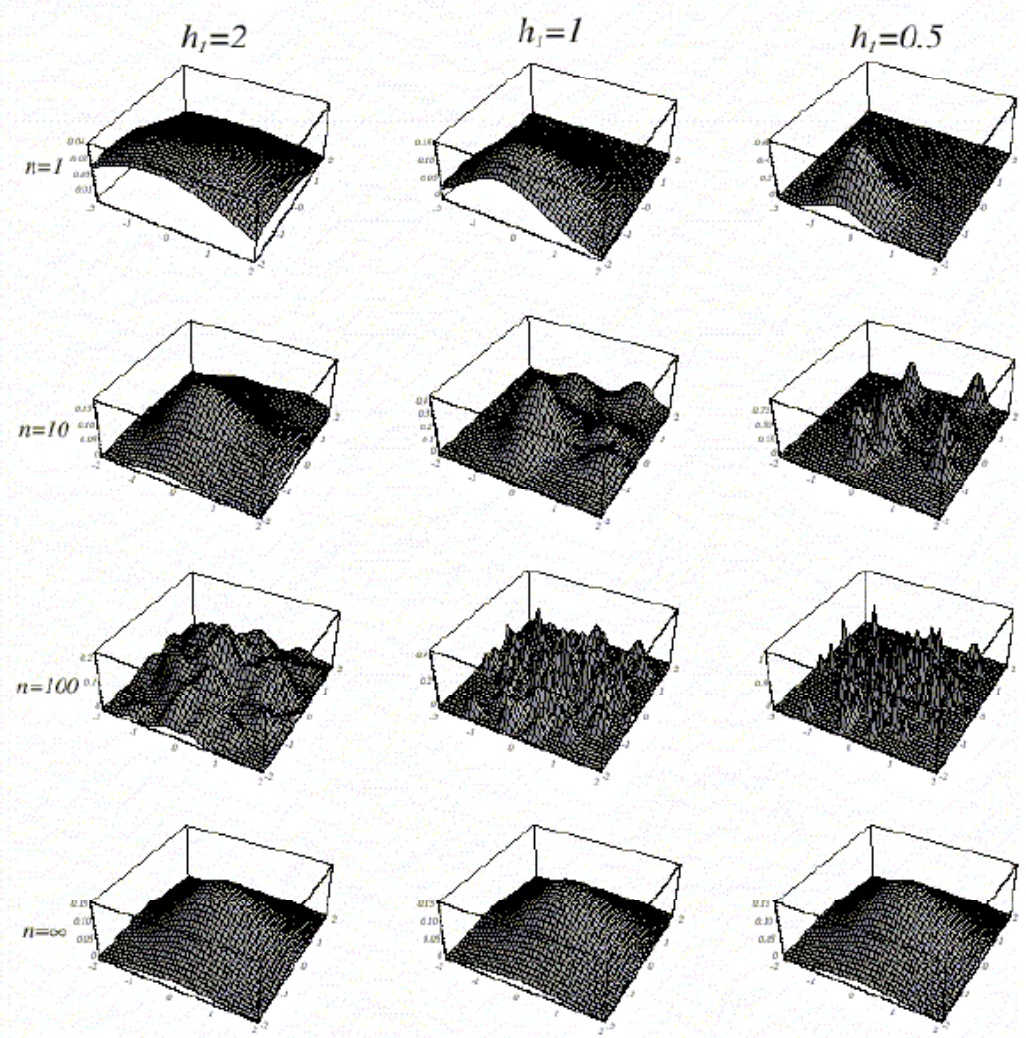
Καθώς το *n* τείνει στο άπειρο, η εκτίμηση γίνεται ακριβής, ανεξάρτητα από το μήκος του παραθύρου.

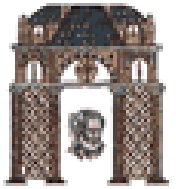


Παράδειγμα παραθύρων Parzen

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n^2}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right);$$

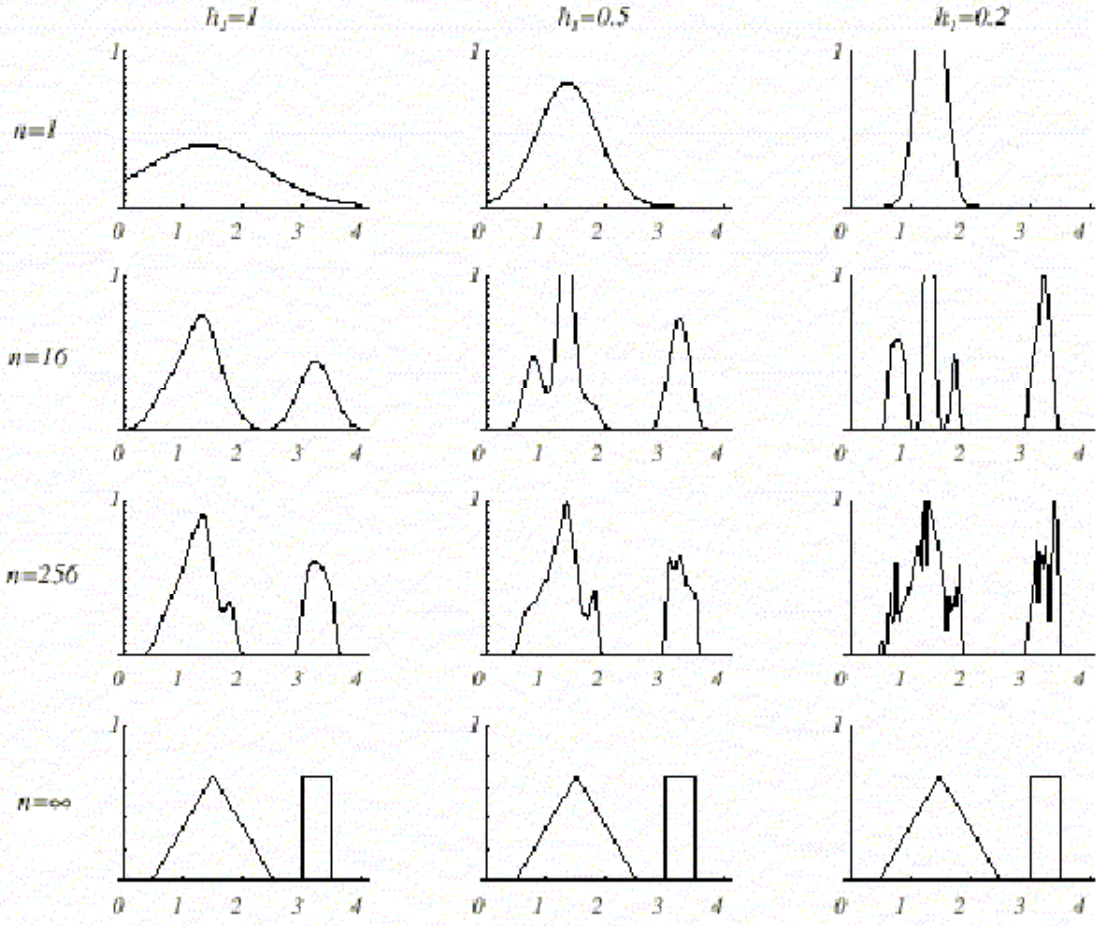
$$\varphi(\mathbf{u}) = \frac{1}{2\pi}e^{\frac{-\mathbf{u}^T\mathbf{u}}{2}}$$

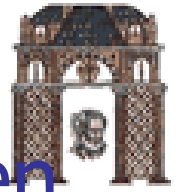




Παράδειγμα παραθύρων Parzen

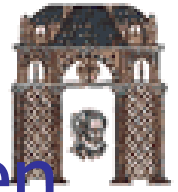
Καθώς το *n* τείνει στο άπειρο, η εκτίμηση γίνεται ακριβής, ανεξάρτητα από το μήκος του παραθύρου.





Ταξινόμηση με χρήση παραθύρων Parzen

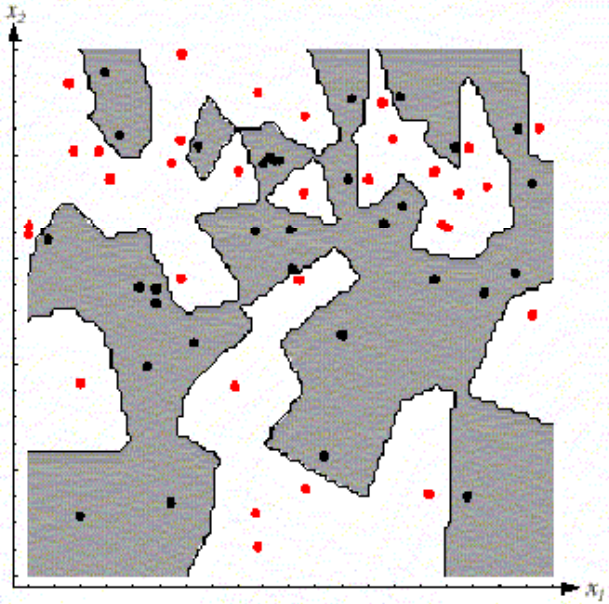
- Εκτίμηση της πιθανοφάνειας $p(\mathbf{x}|\omega_i)$ από τα δεδομένα, χρησιμοποιώντας τη μέθοδο παραθύρων Parzen, και χρήση του κανόνα του Bayes για την ταξινόμηση, δηλ. υπολογισμός των εκ των υστέρων πιθανοτήτων, και επιλογή της κλάσης με την μεγαλύτερη πιθανότητα.
- Πλεονεκτήματα: Δεν προϋποθέτει καμία γνώση για το πρόβλημα, εκτός από την ύπαρξη του συνόλου δειγμάτων εκπαίδευσης!
- Μειονεκτήματα: Απαιτεί $(\text{πολλά})^d$ δεδομένα για να εξασφαλίσει ότι η εκτίμηση συγκλίνει στην πραγματική κατανομή.
  - Επιπλέον, καθώς η διάσταση αυξάνει, η απαίτηση για $(\text{πολλά})^d$ δεδομένα γίνεται $((\text{πολλά})^{\text{πολλά (n)}})^n$!!!! ➔ Πρόβλημα διάστασης (Curse of dimensionality) !
  - Ο μόνος τρόπος για την αντιμετώπιση του είναι η ύπαρξη εκ των προτέρων, σωστής πληροφορίας για τα δεδομένα!
- Το λάθος εκπαίδευσης μπορεί να γίνει αρκούντως μικρό (ακόμα και μηδέν), επιλέγοντας αρκετά μικρά παράθυρα! Παρ' όλα αυτά, δεν είναι επιθυμητό, επειδή σίγουρα θα προκαλέσει overfitting (υπερταίριασμα) και θα μειώσει την απόδοση στα νέα αταξινόμητα δεδομένα ελέγχου (test data).

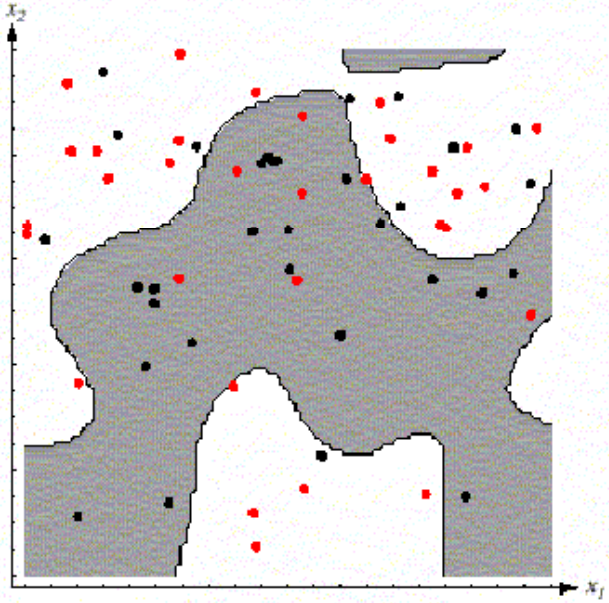


Ταξινόμηση με χρήση παραθύρων Parzen

Πολύ μικρό παράθυρο → Πολύ μικρή διαμέριση του χώρου χαρακτηριστικών, πράγμα μη επιθυμητό!

Μεγαλύτερο παράθυρο → Υψηλότερο λάθος κατά την εκπαίδευση, αλλά καλύτερη απόδοση γενίκευσης! Καλύτερη απόδοση γενίκευσης: Επιθυμητή ιδιότητα.





Στην πράξη, αυτό που θα θέλαμε είναι παράθυρα μικρού πλάτους στις περιοχές με υψηλή πυκνότητα δεδομένων, και παράθυρα μεγάλου πλάτους στις περιοχές όπου τα δεδομένα είναι αραιά! Πώς μπορεί να επιτευχθεί αυτό...?

Πιθανοτικά Νευρωνικά Δίκτυα
Probabilistic Neural Networks (PNN)

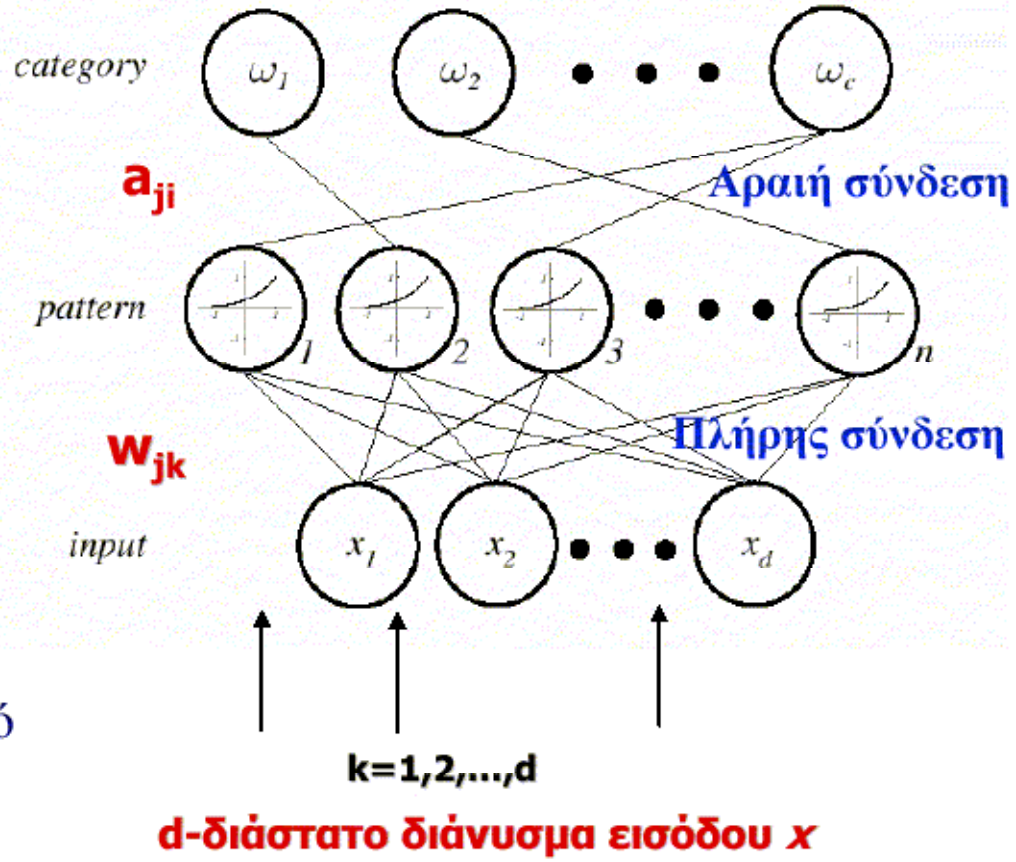
- Είσοδος: $\{x_k;\ k=1,\ldots,d\}$ d κόμβοι, καθένας αντιστοιχεί σε ένα χαρακτηριστικό.

- w_{jk}: βάρη που συνδέουν την k–στή είσοδο με τον j–στό κόμβο κρυφού επιπέδου (κόμβο προτύπου).

- Κρυφό επίπεδο: n κόμβοι, καθένας αντιστοιχεί σε ένα πρότυπο, δηλαδή δείγμα εκπαίδευσης, $j=1,2,\ldots,n$.

- Επίπεδο εξόδου: c κόμβοι, καθένας παριστά μια κλάση.

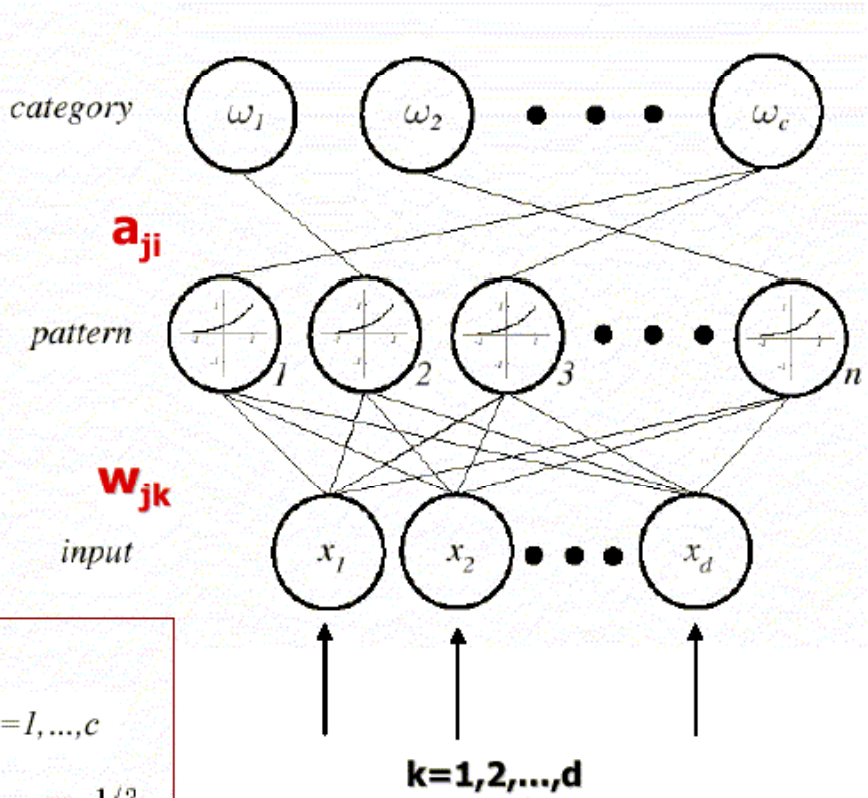
- a_{ji}: βάρη που συνδέουν j–στό κρυφό κόμβο με τον i–στό κόμβο εξόδου, $i=1,2,\ldots,c$



PNN - Εκπαίδευση

Εκπαίδευση

- Το *j-στό* δείγμα εκπαίδευσης (πρότυπο) κανονικοποιείται να έχει μέτρο μονάδα.
- Τοποθετείται στους κόμβους εισόδου.
- Τα βάρη w_{jk} ορίζονται ως $w_{jk}=x_{jk}$.
- Μία μοναδική σύνδεση με βάρος $a_{ji}=1$ γίνεται από τον πρώτο κρυφό κόμβο σε εκείνο τον κόμβο του επιπέδου εξόδου που αντιστοιχεί στην (γνωστή) κλάση του $\mathbf{x}_j$.

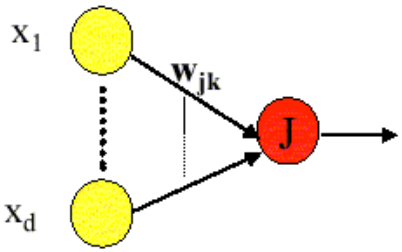


Algorithm 1 (PNN training)

1 **begin initialize** $j=0,\ a_{ji}=0$ for $j=1,...,n;\ i=1,...,c$
2 **do** $j \leftarrow j+1$
3 normalize : $x_{jk} \leftarrow x_{jk} / \left(\sum_i^d x_{ji}^2\right)^{1/2}$
4 train : $w_{jk} \leftarrow x_{jk}$
5 **if** $x \in \omega_i$ **then** $a_{ji} \leftarrow 1$
6 **until** $j = n$

PNN – Ταξινόμηση

- Κάθε κόμβος προτύπου δημιουργεί το εσωτερικό γινόμενο του διανύσματος βαρών του και της κανονικοποιημένης εισόδου ***x*** για να υπολογίσει το $net_J = \mathbf{w}^t\mathbf{x}$, και να αποδώσει $e^{[(netJ-1)/\sigma 2]}$.

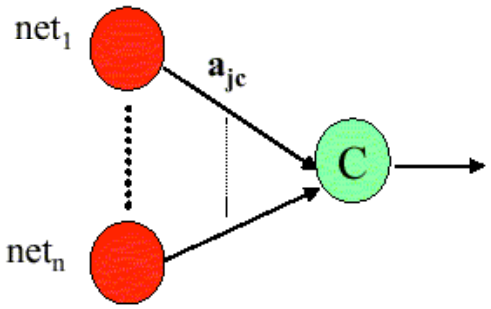


$$e^{\left((net_J - 1)/\sigma^2\right)}$$

$$\varphi\left(\frac{\mathbf{x} - \mathbf{w}_k}{h_n}\right) \propto \overbrace{e^{-(\mathbf{x}-\mathbf{w}_k)^t(\mathbf{x}-\mathbf{w}_k)/2\sigma^2}}^{\text{desired Gaussian}} = e^{-(\mathbf{x}^t\mathbf{x}+\mathbf{w}_k^t\mathbf{w}_k-2\mathbf{x}^t\mathbf{w}_k)/2\sigma^2} = \underbrace{e^{(z_k-1)/\sigma^2}}_{\text{Συνάρτηση ενεργοποίησης}}$$

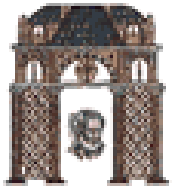
Πλάτος Γκαουσσιανού παραθύρου Parzen

- Κάθε κόμβος κλάσης αθροίζει τα αποτελέσματα των κόμβων προτύπων που συνδέονται με αυτόν. Αυτό εξασφαλίζει ότι η ενεργοποίηση κάθε κλάσης παριστά την εκτίμηση σ.π.π. με κυκλικά συμμετρικό Gaussian παράθυρο Parzen με πίνακα συνδιασποράς $\sigma^2\mathbf{I}_{d\times d}$, όπου ***I*** είναι ο μοναδιαίος πίνακας.



Algorithm 2 (PNN classification)

```
1 begin initialize k = 0, x = test pattern
2       do k ← k + 1
3          z_k ← w_k^t x
4          if a_ki = 1 then g_i ← g_i + exp[(z_k − 1)/σ^2]
5       until k = n
6    return class ← arg max_i g_i(x)
7 end
```

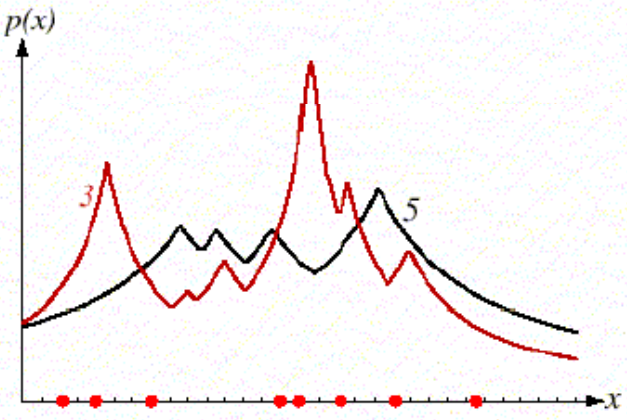


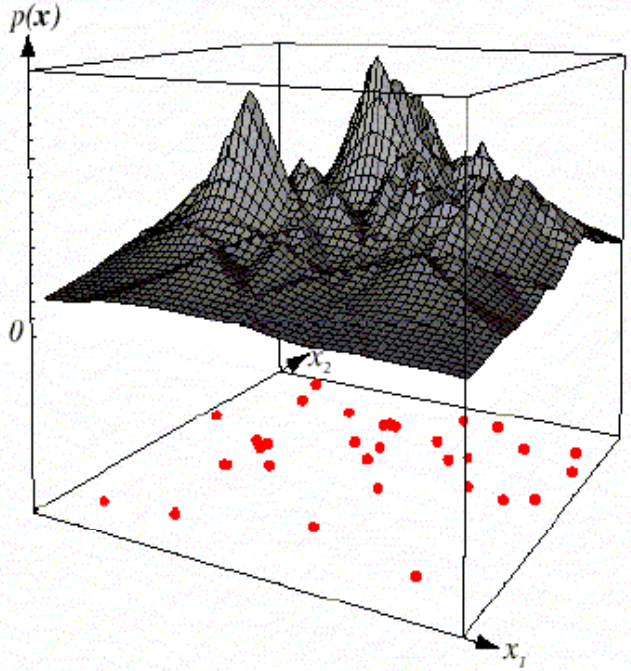
Εκτίμηση k_n-πλησιέστερων γειτόνων
k_n-Nearest neighbour (KNN)

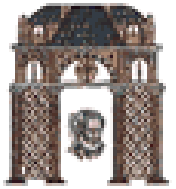
- Αντί να επιλέγουμε το πλάτος του παραθύρου σαν συνάρτηση του πλήθους δειγμάτων ($V_n = V_1 / \sqrt{n}$), γιατί να μην το επιλέγουμε ως μία συνάρτηση των δεδομένων εκπαίδευσης;
- Θυμηθείτε ότι θα θέλαμε ένα μεγάλο παράθυρο στις περιοχές με λίγα δεδομένα, και ένα πιο στενό παράθυρο όπου είναι πυκνή η παρουσία δεδομένων!
- Αλγόριθμος εκτίμησης k-πλησιέστερων γειτόνων:
  - Επιλέγουμε μια αρχική περιοχή γύρω από το $\boldsymbol{x}$ όπου θα θέλαμε να υπολογίσουμε την $p(\boldsymbol{x})$
  - Αυξάνουμε το παράθυρο μέχρι ένα προκαθορισμένο πλήθος k_n δειγμάτων να περιληφθεί εντός του παραθύρου. Αυτοί είναι οι k_n πλησιέστεροι γείτονες του $\boldsymbol{x}$.
  - Υπολογίζουμε την πυκνότητα με βάση την τιμή $\dfrac{k_n/n}{V_n}$
  - Αναγκαίες και ικανές συνθήκες για τη σύγκλιση της $p_n(x)$: $\lim_{n\to\infty} k_n = \infty$; $\lim_{n\to\infty} k_n/n = 0$

KNN

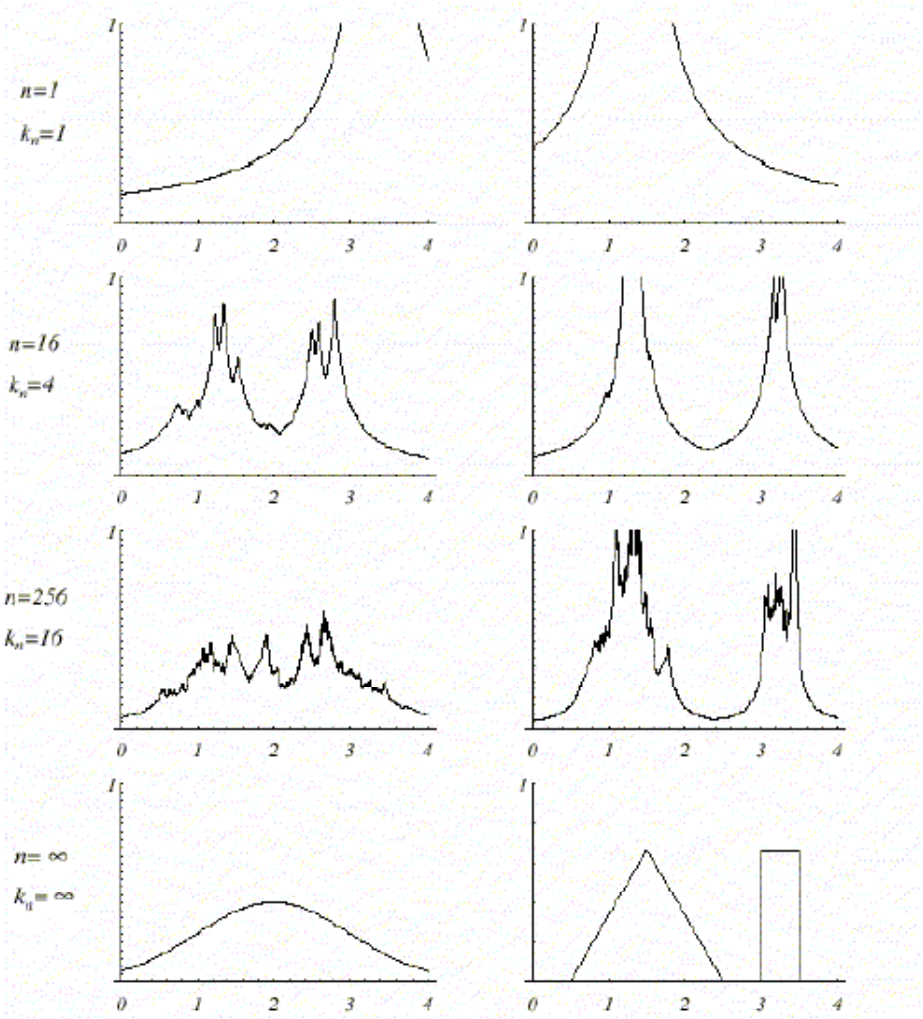
Αν επιλέξουμε $k_n = \sqrt{n}$ και υποθέσουμε ότι η $p_n(\mathbf{x})$ αποτελεί μία αρκετά καλή προσέγγιση της $p(\mathbf{x})$, τότε $V_n \approx 1/\left(\sqrt{n}\,p(\mathbf{x})\right)$. Επομένως, η περιοχή V_n έχει πάλι τη μορφή $V_1/\sqrt{n}$ όπου όμως η αρχική περιοχή V_1 καθορίζεται από την σ.π.π. $p(\mathbf{x})$ των δεδομένων και δεν αποτελεί μια αυθαίρετη επιλογή. Επίσης, για κάθε n, το μέγεθος της περιοχής V_n είναι συνάρτηση του $\mathbf{x}$, δηλ. $V_n = V_n(\mathbf{x})$.





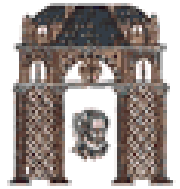


Πώς να επιλέξουμε το k_n



Προσέξτε ότι καθώς το k_n αυξάνει, αυξάνει και η ακρίβεια της εκτίμησης…!

Συνήθως στα προβλήματα ταξινόμησης, προσαρμόζουμε το k_n (ή το h_n για τα Parzen windows), μέχρι ο ταξινομητής να δώσει το χαμηλότερο λάθος για το σύνολο αξιολόγησης (validation test dataset).

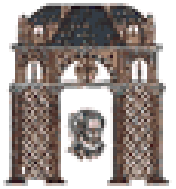


Ταξινόμηση με βάση τον KNN

- Ο KNN μπορεί να χρησιμοποιηθεί για να εκτιμήσει τις εκ των υστέρων πιθανότητες: Στην πραγματικότητα, οι εκ των υστέρων πιθανότητες σε κάθε μικρή περιοχή του $\boldsymbol{x}$ είναι το ποσοστό των δειγμάτων εντός της περιοχής που έχουν ετικέτα ω_i.
  - n: συνολικός αριθμός προτύπων όλων των κλάσεων
  - k_i: αριθμός προτύπων της κλάσης i στην περιοχή γύρω από το $\boldsymbol{x}$
  - k: συνολικός αριθμός προτύπων όλων των κλάσεων στην περιοχή γύρω από το $\boldsymbol{x}$

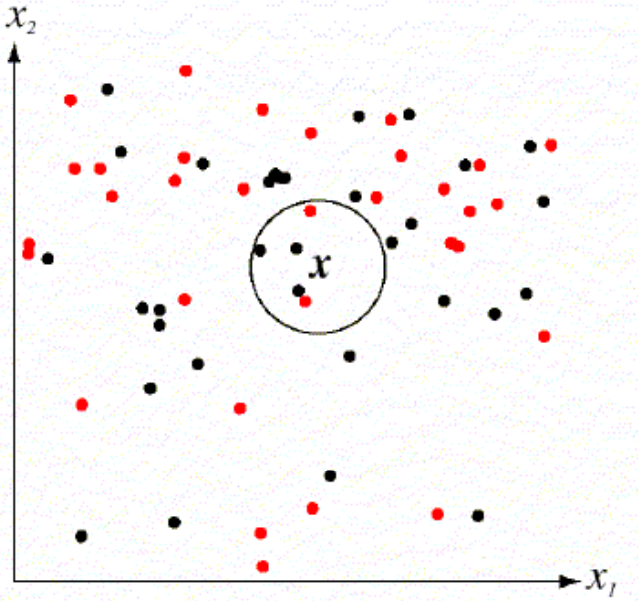
$$p_n(\mathbf{x}, \omega_i) = \frac{k_i / n}{V}$$

$$P_n(\omega_i \mid \mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k}$$



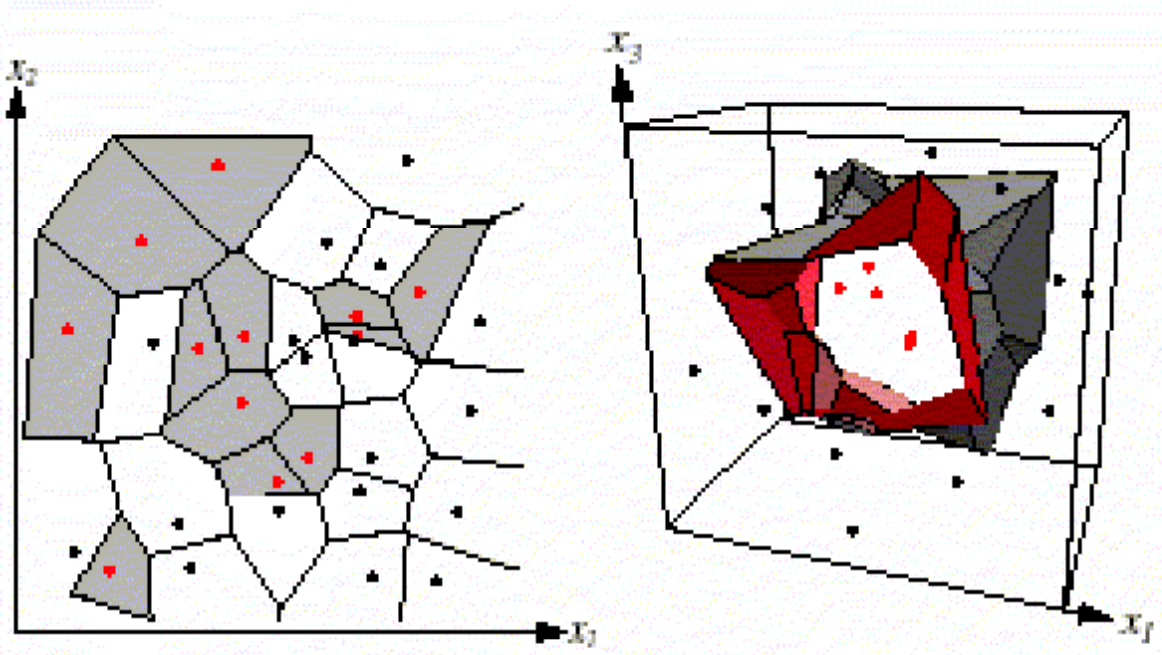
Ταξινομητής KNN

- Για την ταξινόμηση ενός δεδομένου δείγματος $\boldsymbol{x}$,
  - Μεταξύ των n διανυσμάτων εκπαίδευσης, προσδιορίζουμε τους k πλησιέστερους γείτονες του ανεξάρτητα από την κλάση στην οποία ανήκουν, (όπου το k είναι περιττός για ταξινόμηση σε μία από δύο κλάσεις).
  - Προσδιορίζουμε πόσα από δείγματα (έστω k_i) ανήκουν στην τάξη i, $\sum_i k_i = k$
  - Ταξινομούμε το $\boldsymbol{x}$ στην κλάση με το μεγαλύτερο πλήθος k_i δειγμάτων!

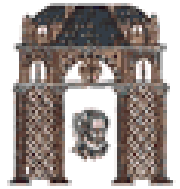


Ταξινομητής Πλησιέστερου Γείτονα

- Ο πιο απλός ταξινομητής KNN είναι αυτός με $k=1$! Αυτός ο ταξινομητής αντιστοιχίζει το x στην τάξη του πλησιέστερου γείτονά του. Είναι καλός αυτός ο ταξινομητής…?



- **Ο NN ταξινομητής οδηγεί στο διαχωρισμό του χώρου ως ενός μωσαϊκού Voronoi, όπου κάθε κελί παίρνει την ετικέτα της κλάσης την οποία περιέχει.**
- **Δεδομένου απείρου αριθμού δειγμάτων εκπαίδευσης, η πιθανότητα λάθους ταξινόμησης έχει ως πάνω όριο, το διπλάσιο της πιθανότητας σφάλματος του Μπεϋζιανού ταξινομητή (ισχύει για μικρές πιθανότητες λάθους).**



Μείωση Υπολογιστικού Φόρτου

- Υπολογισμός μερικών αποστάσεων (partial distance)
  - Χρήση ενός υποσυνόλου r από τις d διαστάσεις, για τον υπολογισμό της απόστασης του δείγματος προς ταξινόμηση από τα δείγματα εκπαίδευσης:

$$D_r(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{r} (a_k - b_k)^2 \right)^{1/2}$$

- Χρήση δένδρων (search tree)
  - Δημιουργία δένδρων όπου τα δείγματα εκπαίδευσης (πρότυπα) συνδέονται επιλεκτικά έτσι ώστε για την ταξινόμηση νέου δείγματος, να απαιτείται ο υπολογισμός της απόστασής του από ορισμένα κομβικά πρότυπα (entry or root) και τα συνδεδεμένα πρότυπα αυτών.
- Διαγραφή ή περικοπή (editing, pruning, or condensing)
  - Διαγραφή προτύπων που περιβάλλονται από πρότυπα της ίδιας κλάσης.
    1. Κατασκεύασε το διάγραμμα Voronoi των αρχικών προτύπων
    2. Για κάθε πρότυπο, αν κάποιος γείτονάς του δεν ανήκει στην ίδια κλάση με αυτό, τσέκαρέ το.
    3. Διέγραψε τα μή τσεκαρισμένα πρότυπα και κατασκεύασε το νέο διάγραμμα Voronoi

ΜΕΙΩΣΗ ΔΙΑΣΤΑΣΕΩΝ: PCA

Principal Component Analysis

- Έστω **x** σε D-διαστάσεις. Θέλουμε κάποιο μετασχηματισμό **A** να μας δώσει ένα **y=wx** σε N-διαστάσεις, ώστε να να μπορούμε να αναπαραστήσουμε ικανοποιητικά το y σε λιγότερες διαστάσεις

Principal component analysis (PCA) is one of the most popular techniques for dimensionality reduction. Starting from an original set of l samples (features), which form the elements of a vector $x \in \mathcal{R}^l$, the goal is to apply a linear transformation to obtain a new set of samples:

$$y = A^T x$$

so that the components of y are uncorrelated. In a second stage, one chooses the most significant of these components. The steps are summarized here:

1. Estimate the covariance matrix S. Usually the mean value is assumed to be zero, $E[x] = 0$. In this case, the covariance and autocorrelation matrices coincide, $R = E[xx^T] = S$. If this is not the case, we subtract the mean. Recall that, given N feature vectors, $x_i \in \mathcal{R}^l$, $i = 1, 2, \ldots, N$, the autocorrelation matrix estimate is given by

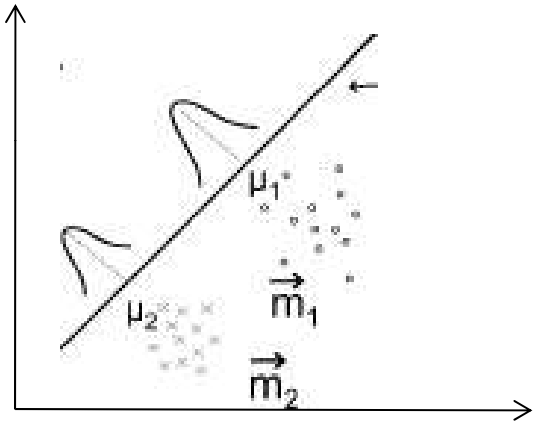
$$R \approx \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T \tag{3.1}$$

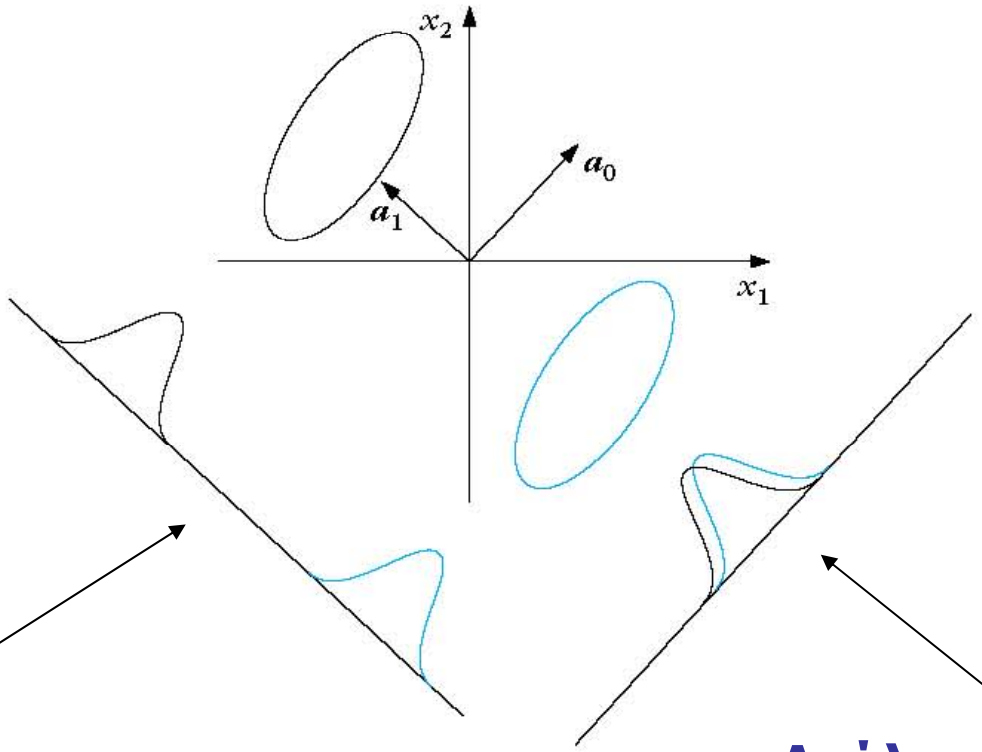
2. Perform the eigendecomposition of S and compute the l eigenvalues/eigenvectors, λ_i, $a_i \in \mathcal{R}^l$, $i = 0, 2, \ldots, l-1$.

3. Arrange the eigenvalues in descending order, $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{l-1}$.
4. Choose the m largest eigenvalues. Usually m is chosen so that the gap between λ_{m-1} and λ_m is large. Eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{m-1}$ are known as the *m principal components*.
5. Use the respective (column) eigenvectors a_i, $i = 0, 1, 2, \ldots, m-1$ to form the transformation matrix

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{m-1} \end{bmatrix}$$

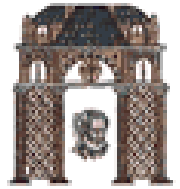
6. Transform each l-dimensional vector x in the original space to an m-dimensional vector y via the transformation $y = A^T x$. In other words, the ith element $y(i)$ of y is the *projection* of x on a_i $\left(y(i) = a_i^T x\right)$.





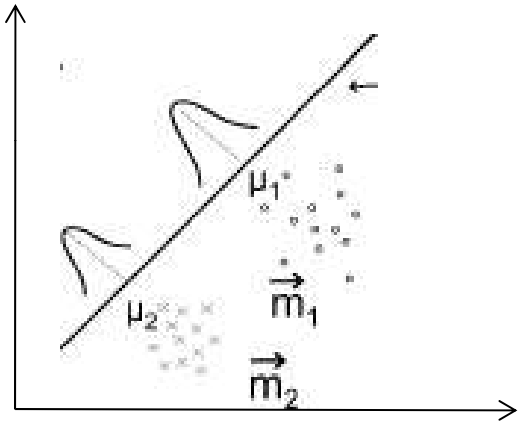
Ανάλυση με κριτήριο την μέγιστη διαχωριστικότητα (Fisher Eye)

Ανάλυση σε κύριες συνιστώσες (PCA)



ΜΕΙΩΣΗ ΔΙΑΣΤΑΣΕΩΝ: Τεχνική Fisher

- Έστω **y** σε D-διαστάσεις. Θέλουμε κάποιο μετασχηματισμό w να μας δώσει ένα **x=wy** σε N-διαστάσεις, ώστε να κάνουμε την κατηγοριοποίηση σε λιγότερες διαστάσεις



Παράδειγμα D=2 και N=1

- Θέλουμε να διαλέξουμε το μετασχηματισμό w ώστε:
  1. Σε κάθε κατηγορία να έχουμε τα γεγονότα κοντά μεταξύ τους, και
  2. οι κατηγορίες να απέχουν το μέγιστο μεταξύ τους.
- Δηλαδή, θα θέλαμε να ισχύουν:
  1. οι μέσες τιμές να απέχουν το μέγιστο μεταξύ τους και
  2. οι διασπορές των δύο κατηγοριών να είναι μηδέν.

Τα παραπάνω μεταφράζονται ως εξής

$$\left|\mu_1 - \mu_2\right|^2 = \max$$

$$\sigma_1^2 + \sigma_2^2 = \min$$

Και ο Fisher πρότεινε την

$$J(w) = \frac{\left|\mu_1 - \mu_2\right|^2}{\sigma_1^2 + \sigma_2^2} = \max$$

Μετρικές και ΚΝΝ Ταξινόμηση

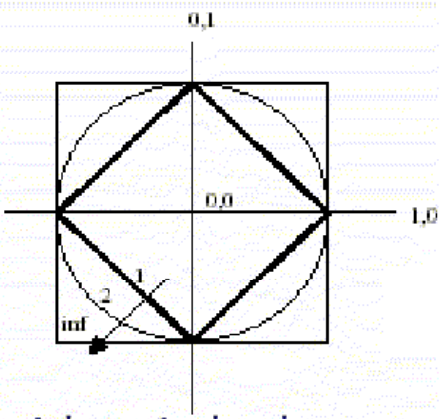
- Ιδιότητες:
  - Nonegativity: $D(\mathbf{a},\mathbf{b}) \geq 0$
  - reflexivity: $D(\mathbf{a},\mathbf{b}) = 0 \quad \text{iff} \quad \mathbf{a} = \mathbf{b}$
  - symmetry: $D(\mathbf{a},\mathbf{b}) = D(\mathbf{b},\mathbf{a})$
  - triangle inequality: $D(\mathbf{a},\mathbf{b}) + D(\mathbf{b},\mathbf{c}) \geq D(\mathbf{a},\mathbf{c})$

- Ευκλείδια Απόσταση: $L_2(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} (a_k - b_k)^2\right)^{1/2}$

- Minkowski Μετρική: $L_p(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} |a_k - b_k|^p\right)^{1/p}$

- Manhatan Απόσταση: ή City Block $L_1(\mathbf{a},\mathbf{b}) = \sum_{k=1}^{d} |a_k - b_k|$

- Chess-board Απόσταση: $L_\infty(\mathbf{a},\mathbf{b}) = \max_{k=1,\ldots,d} (|a_k - b_k|)$



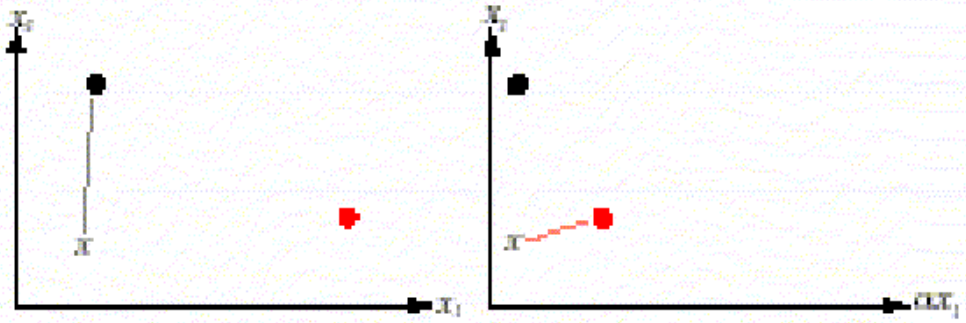
Απόσταση 1 από το κέντρο χρησιμοποιώντας κάθε μία από τις μετρικές L_p

- Mahalanobis Απόσταση σημείου **x** από ένα σύνολο σημείων $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$

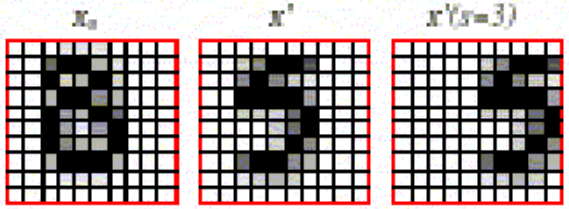
$$L_m(\mathbf{x}, \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

με $\quad \boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n, \quad \Sigma = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu})^T (\mathbf{x}_n - \boldsymbol{\mu})$

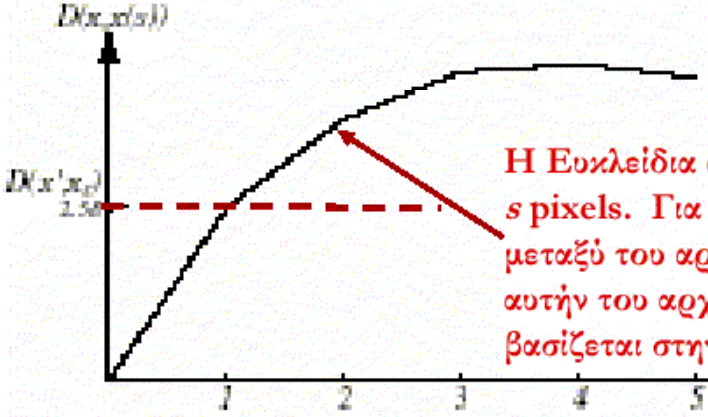
Μετρικές και KNN Ταξινόμηση



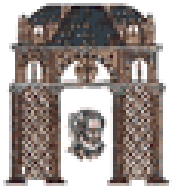
Όταν ο χώρος των προτύπων μετασχηματίζεται πολλαπλασιάζοντας κάθε χαρακτηριστικό με κάποια σταθερά, οι αποστάσεις στο μετασχηματισμένο χώρο μπορεί να είναι σημαντικά διαφορετικές από τις αρχικές αποστάσεις. Προφανώς αυτό επηρεάζει την απόδοση του KNN ταξινομητή.



Είναι σημαντική η εύρεση μετρικών που να μην επηρεάζονται από κάποιους βασικούς μετασχηματισμούς όπως μετατόπιση (shift), κλιμάκωση (scaling), περιστροφή (rotation), πάχος γραμμής (line thickness), στρέβλωση (shear). Παράδειγμα: οπτική αναγνώριση χαρακτήρων (optical character recognition – OCR).



Η Ευκλείδια απόσταση μεταξύ ενός 5 και ενός μετατοπισμένου 5 κατά *s* pixels. Για μετατοπίσεις μεγαλύτερες από 1 pixel, η απόσταση μεταξύ του αρχικού 5 και του μετατοπισμένου 5 είναι μεγαλύτερη από αυτήν του αρχικού 5 και ενός 8, και επομένως ο NN ταξινομητής που βασίζεται στην Ευκλείδια απόσταση πραγματοποιεί λάθος ταξινόμηση.

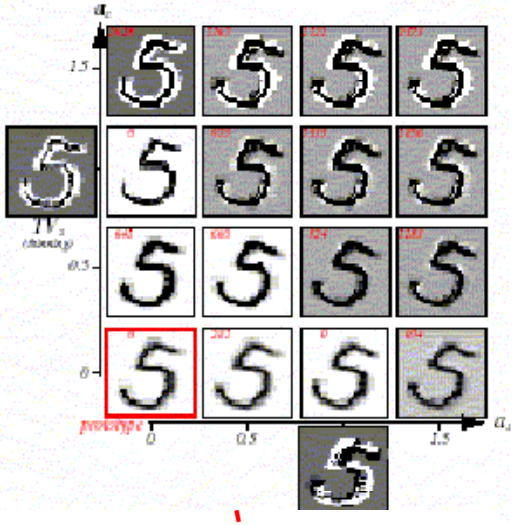


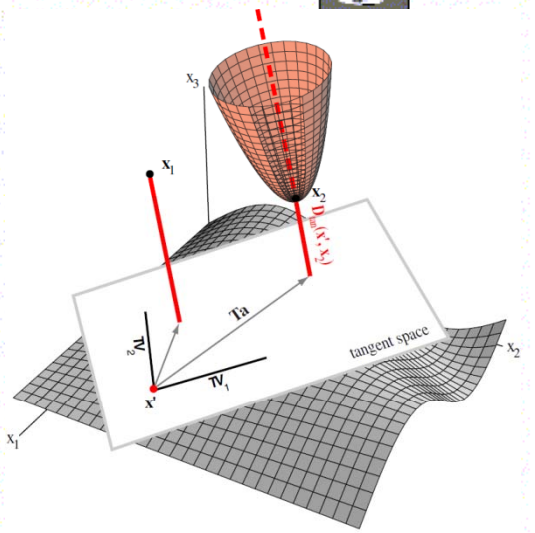
Εφαπτομένη Απόσταση (Tangent Distance)

- Έστω r μετασχηματισμοί, a_i.
- Έστω $\boldsymbol{x}'$ ένα από τα πρότυπα.
- Μετασχηματισμένο πρότυπο, $F_i(\boldsymbol{x}';a_i)$.
- Εφαπτόμενο Διάνυσμα (tangent vector): $\boldsymbol{TV}_i=F_i(\boldsymbol{x}';a_i)-\boldsymbol{x}'$.
- Πίνακας εφαπτομένων (tangent matrix): $\boldsymbol{T}_{dxr}=[\boldsymbol{TV}_1,\ldots,\boldsymbol{TV}_r]$.
- Χώρος εφαπτομένων (tangent space): Ο χώρος που ορίζεται από τα r γραμμικά ανεξάρτητα εφαπτόμενα διανύσματα $\boldsymbol{TV}_i$ που περνούν από το $\boldsymbol{x}'$. Αποτελεί μία γραμμική προσέγγιση του χώρου των μετασχηματισμένων $\boldsymbol{x}'$.
- Εφαπτομένη απόσταση (tangent distance):

$$D_{\tan}(\mathbf{x}',\mathbf{x})=\min_{\mathbf{w}}\left[\left\|\left(\mathbf{x}'+\mathbf{T}\mathbf{w}\right)-\mathbf{x}\right\|\right]$$

- Είναι η ευκλείδια απόσταση του $\boldsymbol{x}$ από το χώρο εφαπτομένων του $\boldsymbol{x}'$





Δίκτυα Μειωμένης Ενέργειας Coulomb
Reduced Coulomb Energy (RCE) Networks

- **Το RCE δίκτυο ρυθμίζει κατά την εκπαίδευση το πλάτος του παραθύρου γύρω από κάθε πρότυπο σύμφωνα με την απόσταση του από το πλησιέστερο πρότυπο μιας διαφορετικής κλάσης.**
- **Κατά την εκπαίδευση, κάθε πρότυπο εισάγει ένα νέο κύκλο και οι ακτίνες των κύκλων προσαρμόζονται ώστε να μην περιέχουν πρότυπα διαφορετικών κλάσεων.**
- **Οι μαύροι κύκλοι παριστούν την κλάση 1, οι ροζ κύκλοι την τάξη 2, ενώ οι σκούρες κόκκινες περιοχές παριστούν ασαφείς περιοχές όπου δεν μπορεί να ληφθεί απόφαση.**