DISCRIMINAZIONE SEGNALE-FONDO IN HEP

S.Giagu

FNSN2 - Roma - 26/28 Maggio 2008

SOMMARIO

- Separazione segnale-fondo in fisica delle alte energie
- Il metodo classico: classificazione basata sui "tagli"
- Cenni di teoria della classificazione statistica:
  - classificatore ottimale
  - misura delle prestazioni di un classificatore
  - algoritmi di ottimizzazione
- Classificatori statistici multivariati:
  - generalità
  - classificatori lineari: PDE, Fisher, ...
  - classificatori non lineari: Neural Networks, BDT, SVM, ...
  - prestazioni, robustezza, velocità, trasparenza, curse of dimensionality
- Esempi: analisi Toy

DOCUMENTAZIONE

- Questioni generali di statistica:
  - G.Cowan: Statistical Data Analysis, Oxford Univ.Press
  - appunti lezioni statistica di G.D'Agostini

- Classificatori multivariati:
  - K.Fukemaga: Introduction to statistical pattern recognition, Academic Press
  - D.W.Scott: Multivariate Density Estimation, J.Wiley&Sons

- TMVA (disponibile dentro Root)
  - http://root.cern.ch/root/doc/RootDoc.html

DISICRIMINAZIONE SEGNALE-FONDO IN HEP

- la possibilita' di classificare gli eventi osservati in classi di appartenenza (ex. segnale/fondo) ricopre un ruolo fondamentale nella maggior parte delle analisi di fisica delle alte energie (HEP):
  - separazione a livello di evento: ex. $H \rightarrow WW$ VS $pp \rightarrow WW$
  - identificazione di particelle: ex. separazione $\pi^{\pm}/K^{\pm}$, γ/π^0, τ/Jet, ...
  - flavor tagging: ex. Jet prodotti da quark pesanti (b e c) VS Jet prodotti da quark leggeri (u,d,s)
  - stima di parametri fisici: ex. violazione di CP, mixing, ...
- concettualmente la procedura utilizzata e' molto simile in tutti i casi:

<u>Informazioni in ingresso:</u>
- variabili cinematiche (m, p, ϑ, ...)
- proprieta' evento (molteplicita' leptoni/ tracce/jets, Σq, ..)
- topologia evento (sfericita', variabili Fox-Wolfram, ...)
- risposta dei vari detector (hits, dE/dx, $\theta_{Cherenkov}$, profili sciami em/had, ...)
- ...

Input

Algoritmo di Classificazione:
- mono/multi dimensionale
- lineare/non-lineare
- risposta binaria/continua

Risposta

- S / B
- Prob(S)

<u>tradizionalmente</u>: poche variabili potenti

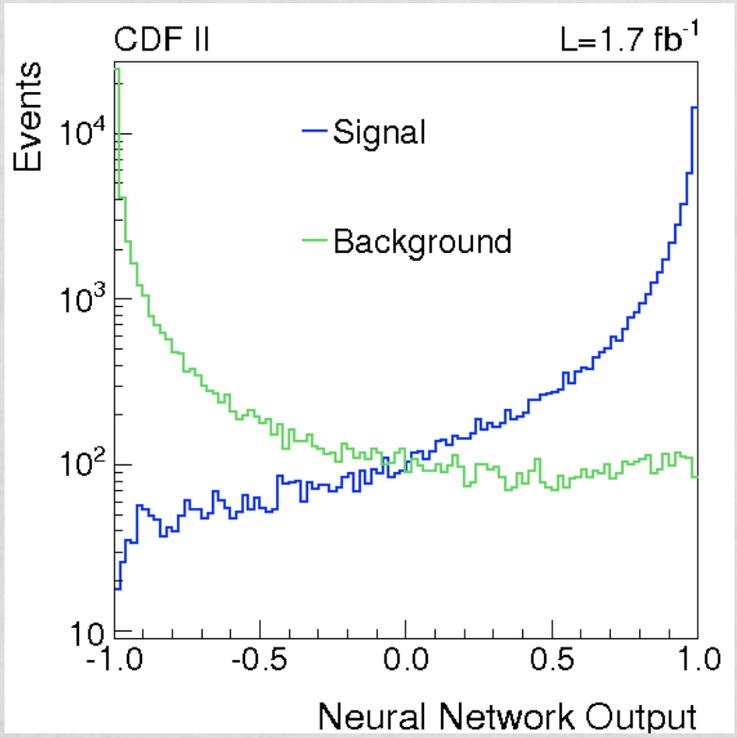
<u>oggi</u>: possibilita' di combinare un grande numero ($O(10^2)$) di variabili diverse senza perdita di potere discriminante

ESEMPIO 1: HF PHYSICS

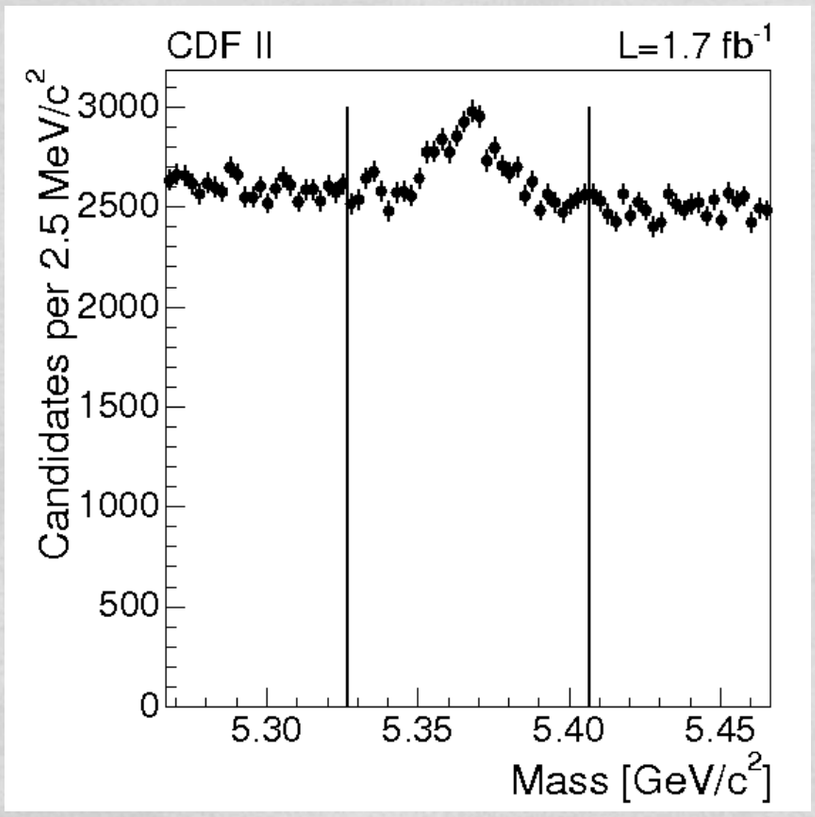
$B_s \rightarrow J/\psi\phi$

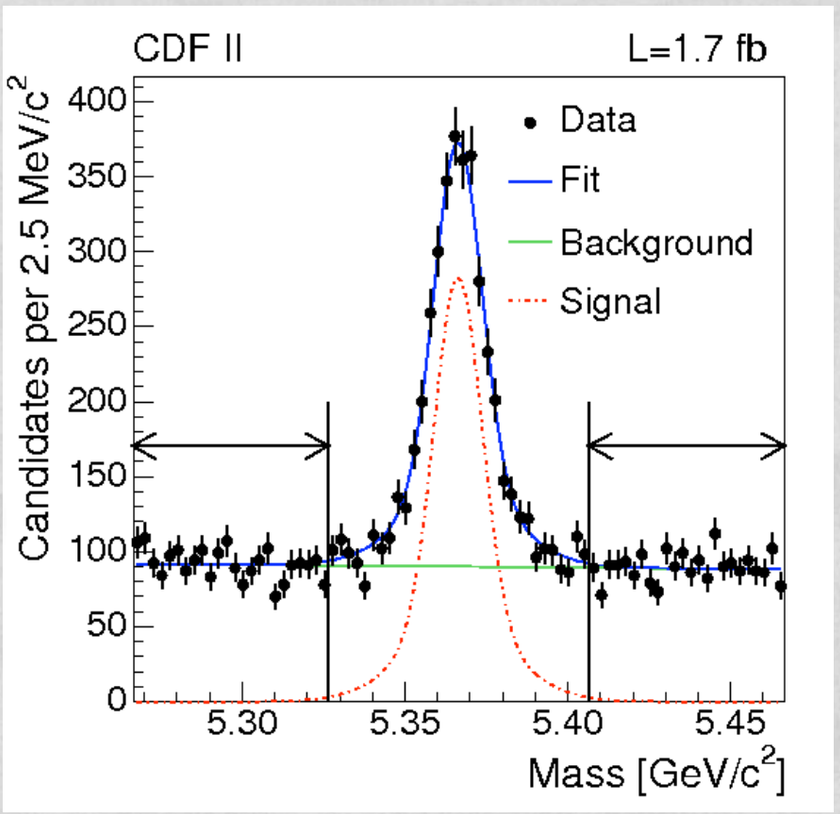
$\phi \rightarrow K^+K^-$

$J/\psi \rightarrow \mu^+\mu^-$



Fondo dominante:
BG combinatoriale



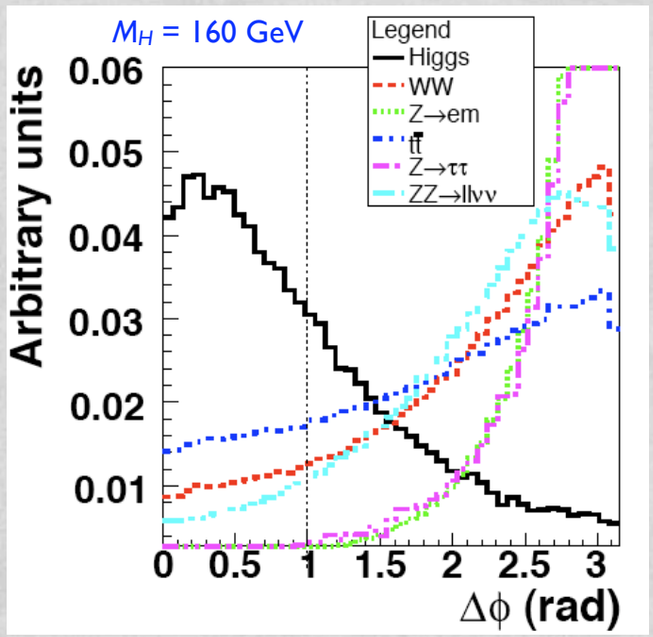


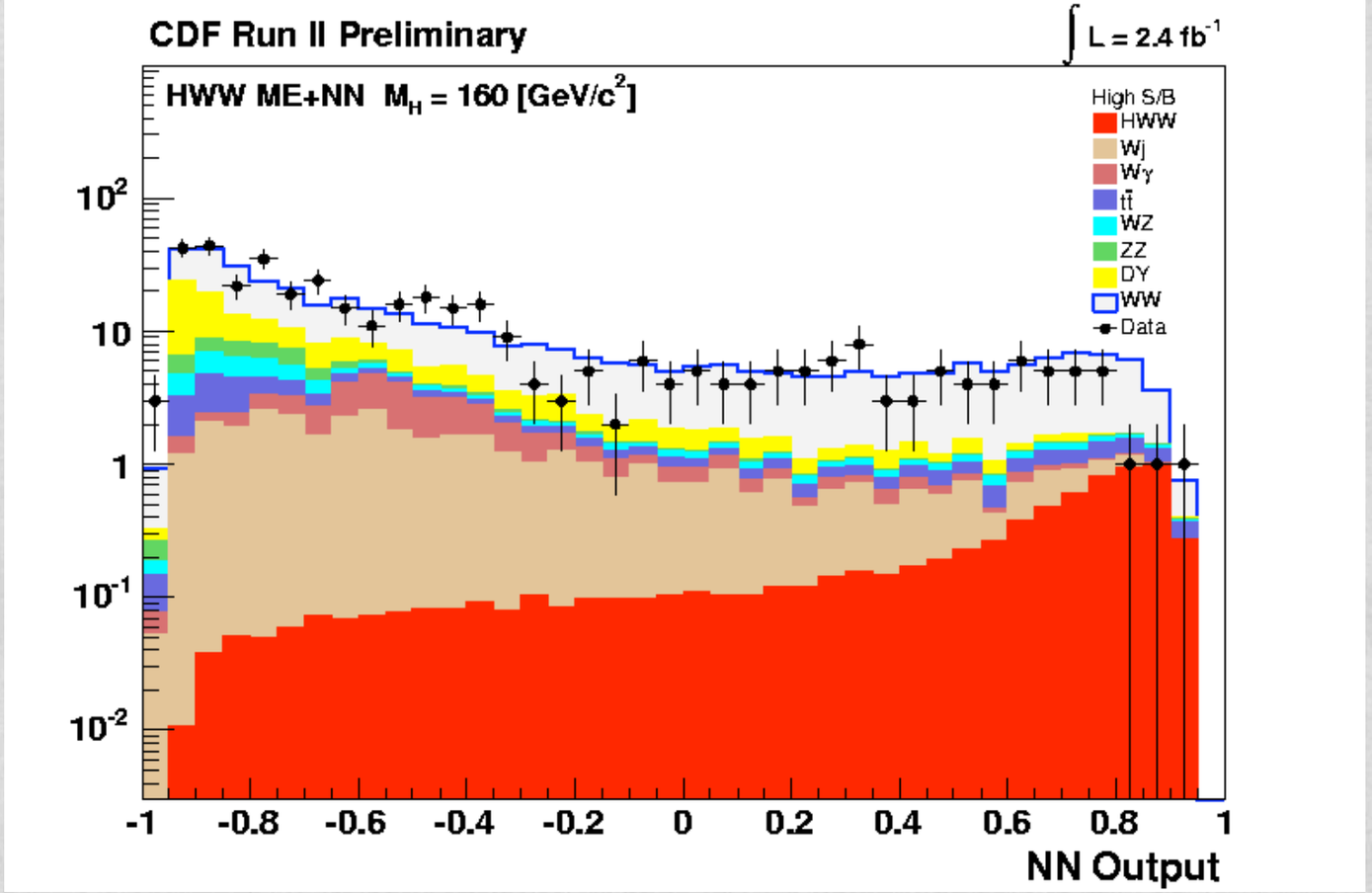
ESEMPIO 2: RICERCA HIGGS

$H \rightarrow W^+W^-$ (↳ $l^+\nu$, ↳ $l^-\nu$)

Fondi dominanti: BG fisici Modello Standard

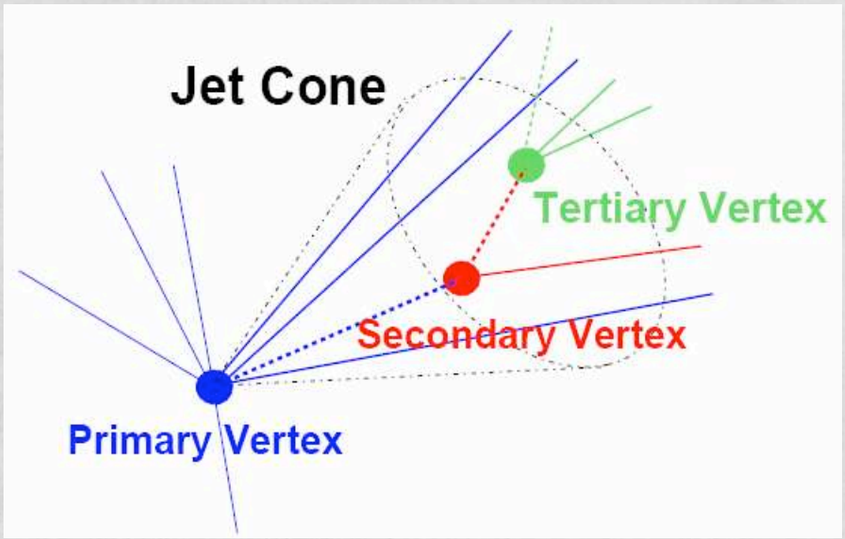
NN
11 variabili in input





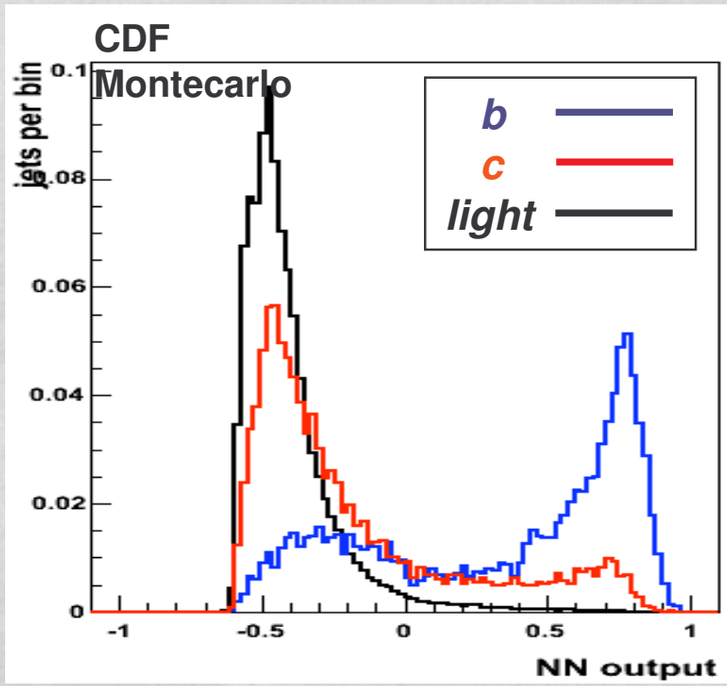
ESEMPIO 3: B-TAGGING

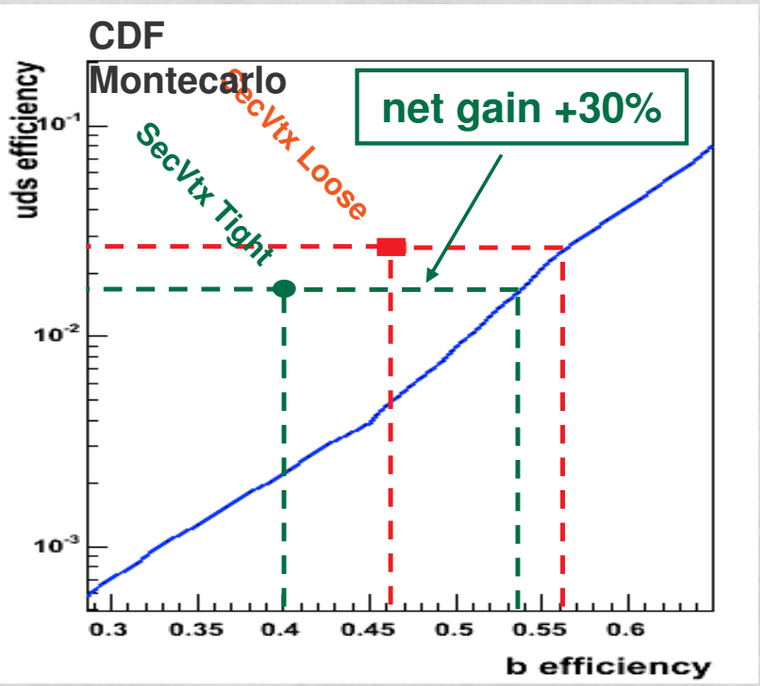
Scopo: distinguere jet adronici prodotti dall'adronizzazione di heavy quark (b e c) da jet da light quark (u,d,s)



caratteristiche peculiari b, c:
- high lifetime
- semileptonic decays (b→clν, c→slν)

Informazioni topologiche e cinematiche combinate insieme tramite cascate di classificatori multivariati: LR/NN



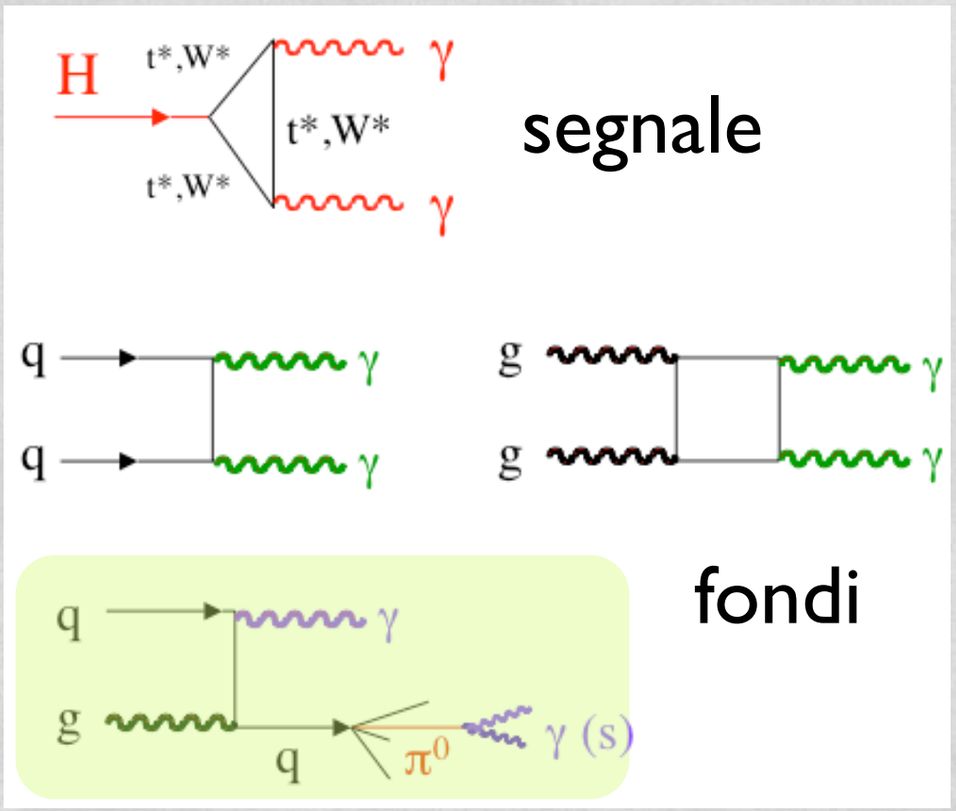


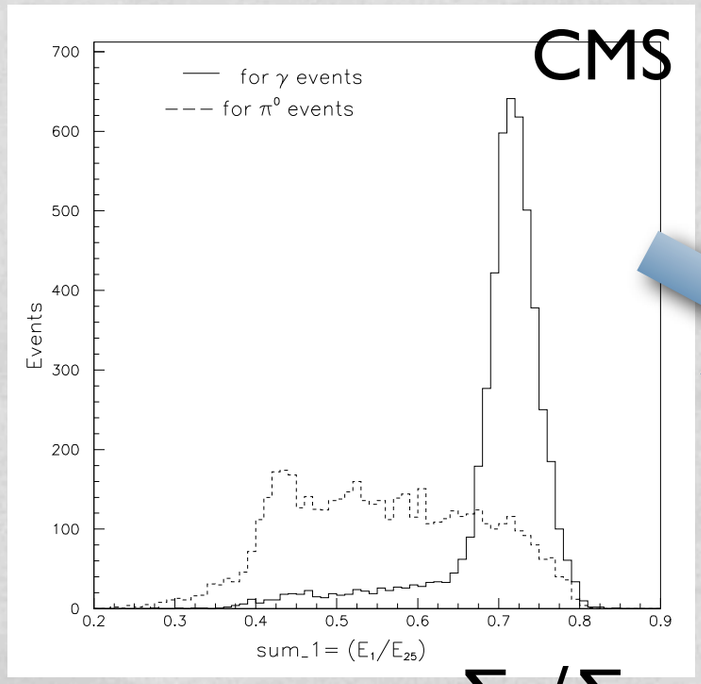
ESEMPIO 4: γ/π^0 REJECTION

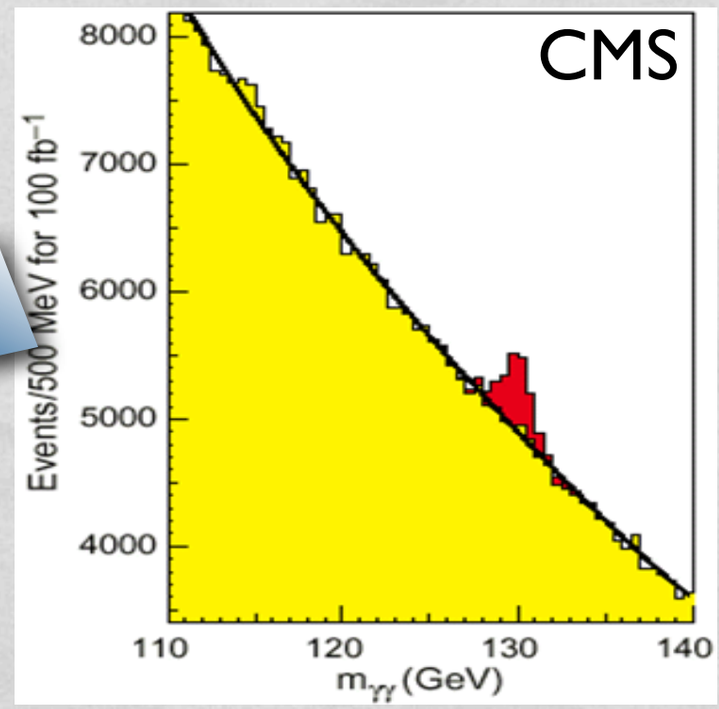
$H \rightarrow \gamma\gamma$

- Canale di ricerca principale a LHC per Higgs di bassa massa
- Misura possibile unicamente se si riesce a "controllare" il fondo riducibile di QCD in cui un π^0 simula un fotone di alta energia nel rivelatore

Utilizzo di classificatori multivariati: algoritmi genetici/ BDT/NN che sfruttano le diverse distribuzioni trasversali/longitudinali dei depositi di energia nei calorimetri e.m.



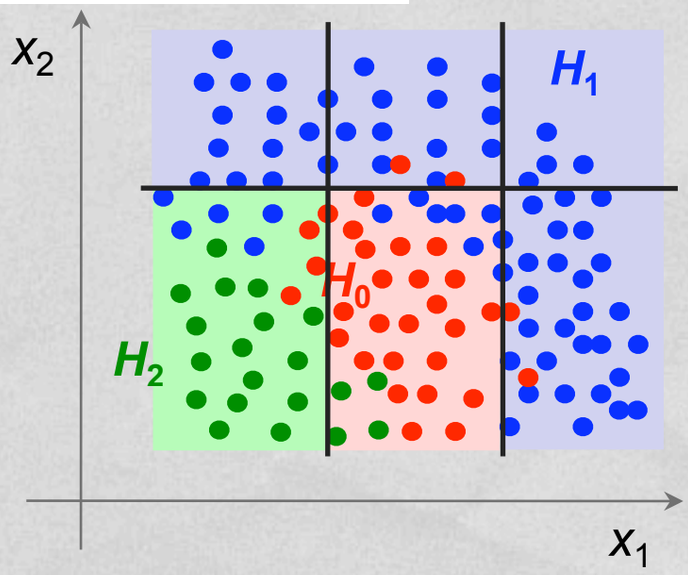




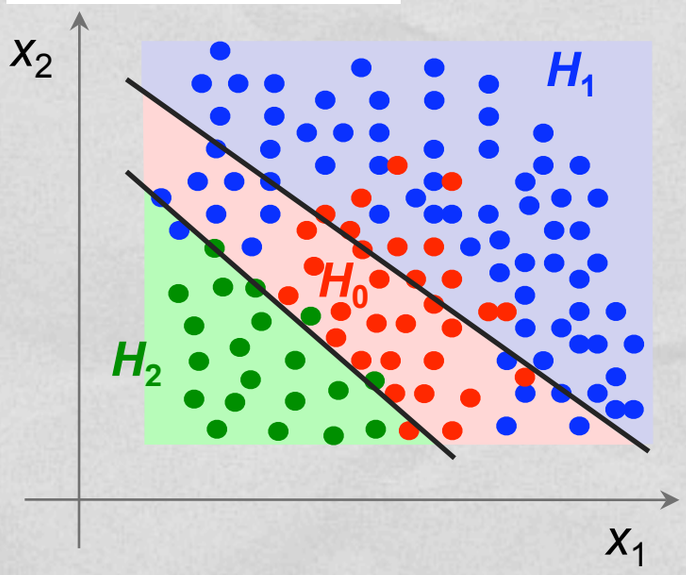
CLASSIFCAZIONE DEGLI EVENTI

- Il problema generale della teoria della classificazione e' quello di determinare a quale categoria o classe appartenga un dato campione
  - attraverso il processo di misura lo sperimentatore ottiene un set di numeri che costituisce il cosidetto vettore di osservabili
  - il vettore di osservabili costituisce l'input per una regola di decisione attraverso la quale il campione viene assegnato ad una di piu' classi H_i (i=1,..,N): Classificatore: $R^n \rightarrow N$ o $R^n \rightarrow R^m$ (m=1 e' il caso piu' usato in HEP)

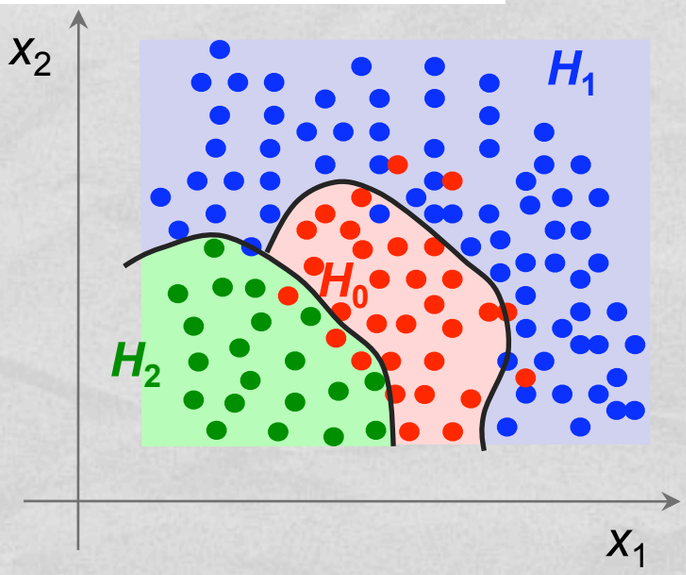
tagli rettangolari



superfici lineari



superfici non lineari

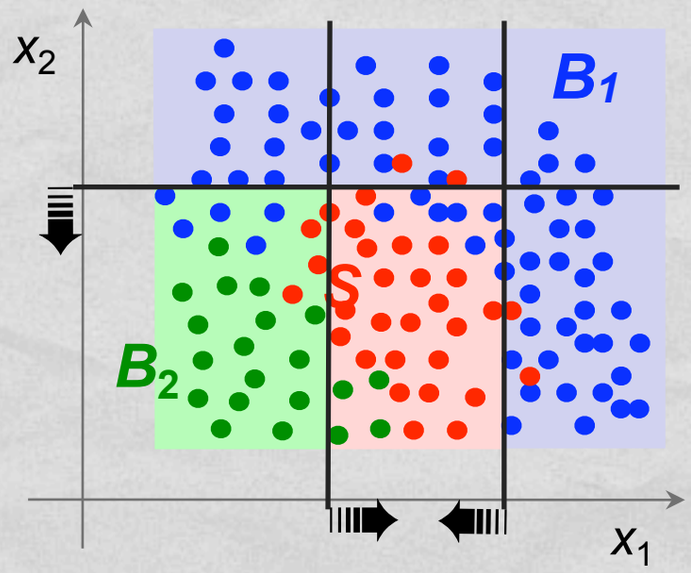


Problema: quale e' il metodo migliore per trovare la superficie di separazione ottimale?

CLASSIFICAZIONE CUT-BASED

- E' il metodo piu' semplice ed utilizzato per separare segnale da fondo
  - trasparente: i.e. semplice da interpretare, dipendenza chiara dall'input
  - risposta binaria: non sfrutta tutta l'informazione disponibile
    - funziona bene quando i campioni sono facilmente separabili
  - difficile da ottimizzare:
    - n=1 → ovvio, n=2 → facile
    - n>2 non ovvio (specie in presenza di correlazioni), non facile (computazionalmente)

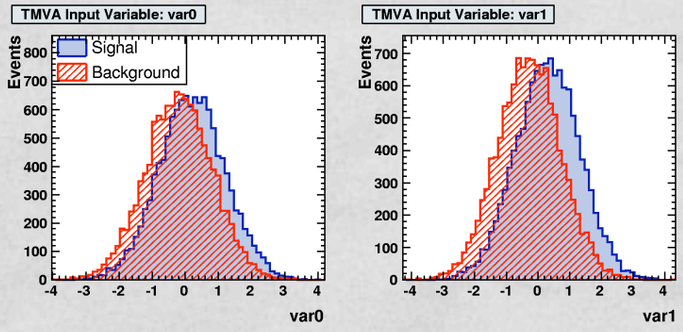
$$x_{\text{cut}}(i_{\text{event}}) \in \{0,1\} = \mathop{\mathrm{I}}_{v \in \{\text{variables}\}} \left(x_v(i_{\text{event}}) \subset [x_{v,\min}, x_{v,\max}] \right)$$

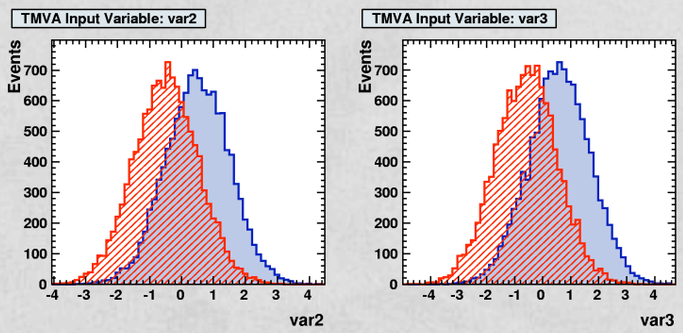


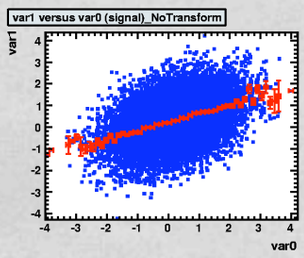
➢ per migliorare le prestazioni: decorrelazione
➢ per accelerare la procedura di ottimizzazione:
MC sampling, algoritmi genetici, simulated annealing,...

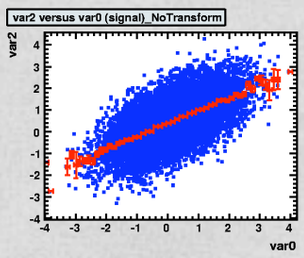
DECORRELAZIONE

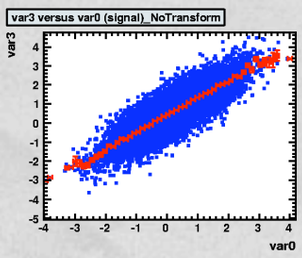
Procedura per rimuovere correlazioni lineari tra le variabili

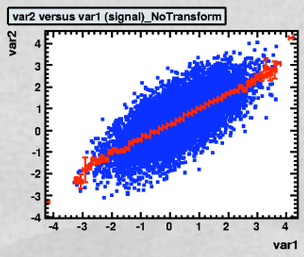


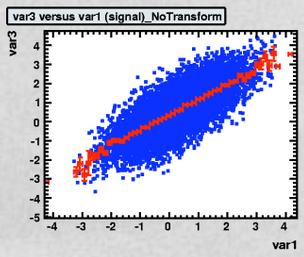


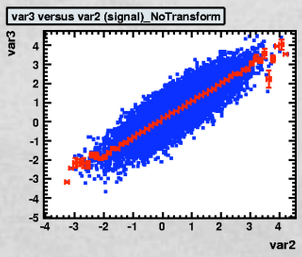






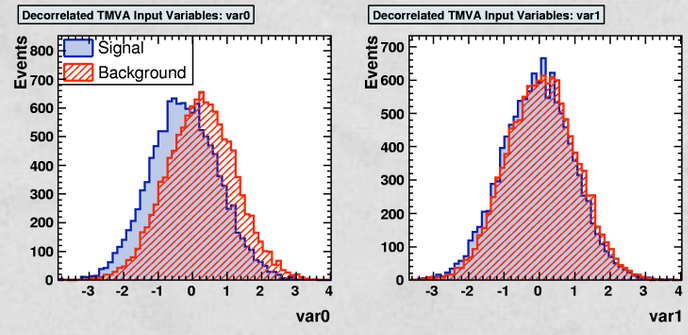


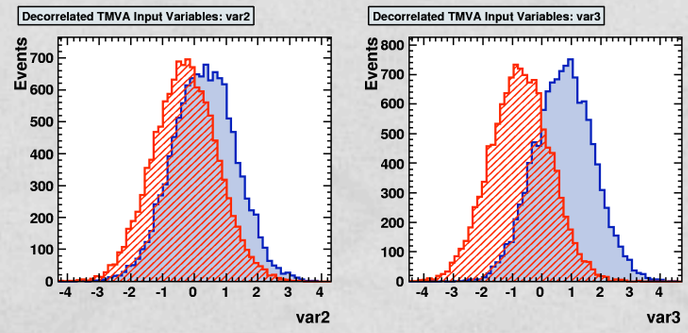


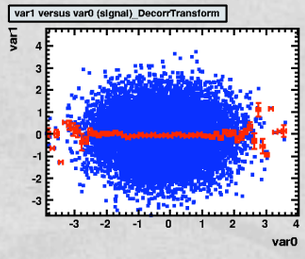


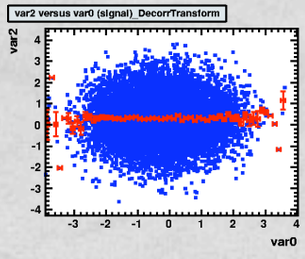
Rotazione delle variabili di input:

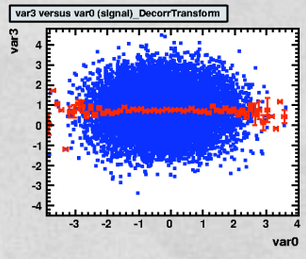
$x' = Ax$

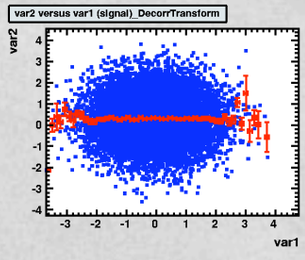


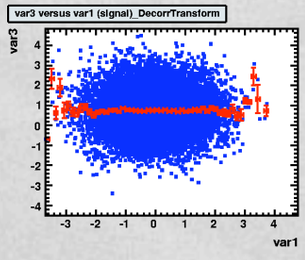


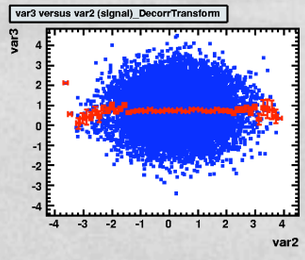












DIAGONALIZZAZIONE SIMULTANEA: TRASFORMAZIONE DI MAHALANOBIS

- Th.: indicate con $\sum_S$ e $\sum_B$ le matrici di covarianza delle classi segnale e fondo, e' sempre possibile costruire una matrice di trasformazione A tale che il vettore di variabili casuali dell'evento trasformato da A: $Y^T=AX^T$ abbia una matrice di covarianza unita' per il campione di eventi in una delle due classi e diagonale nell'altra

- $\sum_B$ simmetrica (hermitiana e reale) $\rightarrow \exists$ decomposizione in autovalori:
  - $\exists$ matrice ortogonale U (reale ed unitaria): $\sum_B = U\Lambda U^T$ e $\Lambda=diag(\lambda_i)$
  - $(\sum_B)^n = U\Lambda^n U^T$ e (autovalori di Λ positivi): $(\sum_B)^{\pm 1/2} = U\Lambda^{\pm 1/2}U^T$
  - Sia ora: $M = (\sum_B)^{-1/2}=U\Lambda^{-1/2}U^T \Rightarrow$ la matrice $M\sum_S M$ e' simmetrica $\Rightarrow$

    $\exists V: M\sum_S M = VDV^T$ con $D=diag(d_i)$
  - Consideriamo la trasformazione: $A = VTM = V^T U\Lambda^{-1/2}U^T$ avremo:

$$
\begin{aligned}
A\Sigma_B A^T &= V^T M \Sigma_B M^T V = V^T U \Lambda^{-1/2} (U^T \Sigma_B U) \Lambda^{-1/2} U^T V \\
&= V^T U \Lambda^{-1/2} \Lambda \Lambda^{-1/2} U^T V = V^T U 1 U^T V = V^T 1 V = V^T V = 1
\end{aligned}
$$

$$
\begin{aligned}
A\Sigma_S A^T &= V^T M \Sigma_S M^T V = V^T (M \Sigma_S M) V = \\
&= V^T (V D V^T) V = (V^T V) D (V^T V) = 1 D 1 = D
\end{aligned}
$$

La trasformazione $A = VTM = V^T U \Lambda^{-1/2} U^T$ permette di utilizzare classificatori che non implementano (ignorano) possibili correlazioni tra variabili: PDE, cut-based, decision trees.

NOTA: il metodo e' in grado di risolvere il problema solo per correlazioni lineari e per variabili distribuite gaussianamente → piu' si e' lontani da tali condizioni piu' la procedura funziona in modo sub-ottimale

metodi alternativi:

- Principal Component Analysis (vedi I.T. Jolliffe, Principal Component Analysis, Series: Springer Series in Statistics, Springer, NY, 2002.)

- Usare variabili non correlate

- Usare classificatori che implementino le correlazioni tra variabili (ex. NN, BDT, ...)

CLASSIFICAZIONE STATISTICA
classificatore ottimale

- Problema dell'assegnazione del risultato di una misura ad una tra due classi di appartenenza:

$$P(S|x) > P(B|x) \rightarrow x \in S$$
$$P(S|x) < P(B|x) \rightarrow x \in B$$

P(L|x) probabilita' a posteriori
che l'evento x appartenga alla classe L

- le probabilita' a posteriori possono essere espresse in termini delle probabilita' a priori tramite il teorema di Bayes:

$$P(l|x) = \frac{P(x|l)P(l)}{P(x)}$$

$$R(x) = \frac{P(x|S)}{P(x|B)} > P(B)/P(S) = \beta \rightarrow x \in S$$

- e' possibile dimostrare in teoria delle ipotesi statistiche come tale regola di decisione risulti ottimale (i.e. minimizza la probabilita' di classificazione errata), una volta scelto opportunamente il valore di soglia β
- NOTA: versione equivalente: con D∈[0,1]

$$D(x) = \frac{P(x|S)}{P(x|S) + P(x|B)} > \alpha \rightarrow x \in S$$

dimostrazione:

supponiamo che vi sia un costo associato ad ogni decisione:

$$c_{11}\ decido\ x \in S\ quando\ x \in S$$
$$c_{12}\ decido\ x \in B\ quando\ x \in S$$
$$c_{21}\ decido\ x \in S\ quando\ x \in B$$
$$c_{22}\ decido\ x \in B\ quando\ x \in B$$

la decisione sbagliata costa di piu' di quella corretta

$$c_{12} > c_{11}$$
$$c_{21} > c_{22}$$

Γ_1, Γ_2: regioni di x per le quali si decida $x \in S$ e $x \in B$

vogliamo scegliere Γ_1, Γ_2 in modo da rendere minimo il valore aspettato del costo (detto rischio r)

$$\begin{aligned} r = E[costo] &= \int_{\Gamma_1} c_{11} p(S|x)p(x)dx + \int_{\Gamma_2} c_{12} p(S|x)p(x)dx + \\ &+ \int_{\Gamma_1} c_{21} p(B|x)p(x)dx + \int_{\Gamma_2} c_{22} p(B|x)p(x)dx = \\ &= \int_{\Gamma_1} [c_{11} p(x|S)p(S) + c_{21} p(x|B)p(B)]dx + \\ &+ \int_{\Gamma_2} [c_{12} p(x|S)p(S)dx + c_{22} p(x|B)p(B)]dx \end{aligned}$$

Γ_1 e Γ_2 non si intersecano e $\Gamma_1 \cup \Gamma_2 = I$:

$$\int_{\Gamma_1} p(x|l)dx = 1 - \int_{\Gamma_2} p(x|l)dx$$

$$
\begin{aligned}
r &= c_{12}p(S) + c_{22}p(B) + \\
&+ \int_{\Gamma_1} [-(c_{12} - c_{11})p(S)p(x|S) + (c_{21} - c_{22})p(B)p(x|B)]dx
\end{aligned}
$$

abbiamo trasformato il problema nello scegliere Γ_1 in modo tale che *r* sia minimo

supponiamo che per un dato *x* l'integrando in *r* sia negativo, possiamo allora diminuire *r* assegnando *x* a Γ_1, nel caso contrario possiamo ridurre *r* assegnando *x* a Γ_2

formalmente:

$$
\begin{aligned}
(c_{21} - c_{22})p(B)p(x|B) < (c_{12} - c_{11})p(S)p(x|S) \Rightarrow x \in \Gamma_1 \\
(c_{21} - c_{22})p(B)p(x|B) > (c_{12} - c_{11})p(S)p(x|S) \Rightarrow x \in \Gamma_2
\end{aligned}
$$

che puo' essere riscritta come:

$$
\begin{aligned}
\frac{p(x|S)}{p(x|B)} &< \frac{(c_{21} - c_{22})}{(c_{12} - c_{11})} \frac{p(B)}{p(S)} \Rightarrow x \in \Gamma_1 \\
\frac{p(x|S)}{p(x|B)} &> \frac{(c_{21} - c_{22})}{(c_{12} - c_{11})} \frac{p(B)}{p(S)} \Rightarrow x \in \Gamma_2
\end{aligned}
$$

NOTA:

$$
\begin{aligned}
(c_{21} - c_{22}) > 0 \\
(c_{12} - c_{11}) > 0
\end{aligned}
$$

per l'assunzione fatta all'inizio

e' equivalente alla regola di Bayes per una funzione di costo simmetrica: c_{12}-c_{11}=c_{21}-c_{22} ← in questo caso il costo diventa la probabilita' di errore.

CVD

POTERE DI SEPARAZIONE

- insieme alla definizione dei criteri di classificazione, e' utile definire uno (o piu') criteri che ne stabiliscano le prestazioni
  - figura di merito nella procedura di ottimizzazione del classificatore quando questo dipenda da "parametri" modificabili
  - permette il confronto tra criteri di classificazione diversi

- campione di eventi in esame costituito da *n* differenti classi di eventi
- ogni classe contribuisce al totale con un frazione f_i: $\sum f_i = 1$
- *x*: osservabile (multidimensionale) che permette di distinguere gli eventi tra le varie classi
- la distribuzione di *x* per il campione in esame e':

$$p_{tot}(x|f) = \sum_{i=1,n} f_i p_i(x)$$

$p_i(x)$ e' la pdf di *x* per eventi di tipo *i*
(NOTA: le p_i sono assunte essere note senza errore)

possiamo utilizzare la risoluzione (errore) sulle f_i come "misura" del potere di separazione dell'osservabile *x* ...

sistematica

SEPARAZIONE STATISTICA

- Vari modi possibili per stimare la risoluzione su f_i:
  - fit di massima verosimiglianza alle *n-1* frazioni, ripetuto su un numero sufficientemente grande di campioni Montecarlo → spread dei risultati
  - principio del Minimum Variance Bound (o Rao-Cramer-Frechet bound):

- dato un qualunque estimatore funzione del set di parametri $\boldsymbol{\theta}=(\theta_1,...,\theta_n)$, esiste un limite alla precisione con cui i vari parameti possono essere stimati:

$$cov(\theta_i, \theta_j) \geq -\left(E\left[\frac{\partial^2 \log L}{\partial\theta_i \partial\theta_j} \right] \right)^{-1}_{ij}$$

L e' la funzione di verosimiglianza

nel caso del nostro problema ($\theta_i \equiv f_i$) il bound assume una forma relativamente semplice:

$$L = \prod_{k=1}^{N} p(x_k|f) \Rightarrow \log L = \sum_{k=1}^{N} \log p(x_k|f)$$

N: dimensione del campione

def. di valore aspettato

$$\begin{aligned}
[cov(f_i, f_j)]^{-1} &= -E\left[\frac{\partial^2 \log L}{\partial f_i \partial f_j}\right] = \\
&= \int \cdots \int -\frac{\partial^2}{\partial f_i \partial f_j} \left(\sum_{k=1}^{N} \log p(x_k|f)\right) \prod_{l=1}^{N} p(x_l|f) dx_l = \\
&= N \int -p(x|f) \frac{\partial^2}{\partial f_i \partial f_j} \log p(x|f) dx
\end{aligned}$$

$$p(x|f) = \sum_{i=1,n} f_i p_i(x) = f_1 p_1(x) + f_2 p_2(x) + \cdots + (1 - f_1 - f_2 - \cdots - f_{n-1}) p_n(x)$$

$$\begin{aligned}
\frac{\partial^2 \log p(x|f)}{\partial f_i \partial f_j} &= \frac{\partial}{\partial f_i} \frac{\log p(x|f)}{\partial f_j} = \frac{\partial}{\partial f_i} \frac{[p_j(x) - p_n(x)]}{p(x|f)} = \\
&= -\frac{[p_j(x) - p_n(x)] \cdot [p_i(x) - p_n(x)]}{[p(x|f)]^2}
\end{aligned}$$

$$[cov(f_i,f_j)]^{-1} = N\int -p(x|f)\times -\frac{[p_j(x)-p_n(x)]\cdot[p_i(x)-p_n(x)]}{[p(x|f)]^2}dx =$$
$$= N\int \frac{[p_j(x)-p_n(x)]\cdot[p_i(x)-p_n(x)]}{p(x|f)}dx \Rightarrow$$

$$[\mathrm{cov}(\mathrm{f}_i,f_j)] = \frac{1}{N}\left[\int \frac{[p_j(x)-p_n(x)]\cdot[p_i(x)-p_n(x)]}{p(x|f)}dx\right]^{-1}_{ij}$$

se siamo interessati al caso n=2: Segnale vs Fondo → solo 1 parametro libero *f*:

$$\sigma^2(f) = \frac{1}{N}\left[\int \frac{[p_S(x)-p_B(x)]^2}{fp_S(x)+(1-f)p_B(x)}dx\right]^{-1}$$

quantita' da minimizzare per ottenere la miglior possibile separazione statistica S-B

NOTA: se S e B sono totalmente separati in x:

$$\sigma^2_{best}(f) = \frac{1}{N}\left[\int \frac{p_S(x)^2}{fp_S(x)}dx - \int \frac{p_B(x)^2}{(1-f)p_B(x)}dx\right]^{-1} = \frac{1}{N}\left[\int \frac{p_S(x)}{f}dx - \int \frac{p_B(x)}{(1-f)}dx\right]^{-1} =$$
$$= \frac{1}{N}\left[\frac{1}{f}-\frac{1}{(1-f)}\right]^{-1} = \frac{f(1-f)}{N}$$

Separation power∈[0,1], indipendente dalla dimensione del campione:

$$SepPower = \frac{\sigma_{best}(f)}{\sigma(f)} = \sqrt{f(1-f)\int \frac{(p_S(x)-p_B(x))^2}{p(x|f)}dx}$$

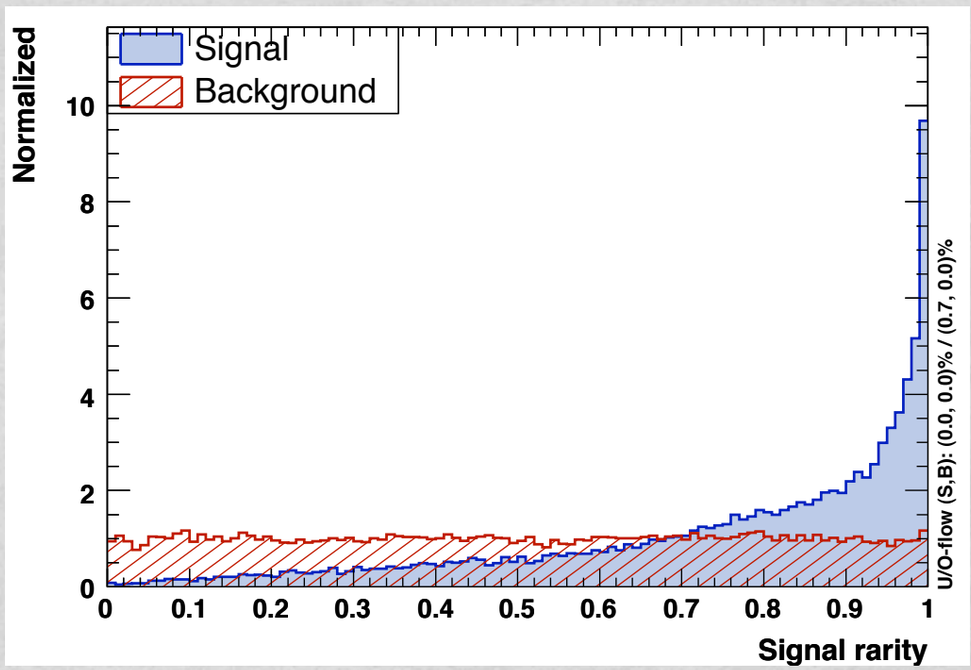
INDICATORI ALTERNATIVI ...

- Esistono metodi alternativi per definire la "potenza" di un dato classificatore, la scelta di uno o dell'altro dipende da considerazioni legate alla semplicita' implementativa e/o interpretativa, generalita' della definizione, precisione della risposta, etc..:
  1. reiezione fondo ad efficienza fissata per il segnale: i.e. $(1-\varepsilon_B)$ VS ε_S
  2. significanza del classificatore: ex. $(<R_S>-<R_B>)/\sqrt{(\sigma^2_{RS}+\sigma^2_{RB})}$, $S/\sqrt{(S+B)}$, ...
  3. rarita':

$$Rarità(y) = \int_{-\inf}^{y} p_B(x)dx$$

B: distribuita uniformemente in [0,1]
S: "clusterizzata" verso y~1

piu' la distribuzione di rarita' e' clusterizzata verso 1 piu' il classificatore risulta discriminante



ALGORITMI DI OTTIMIZZAZIONE

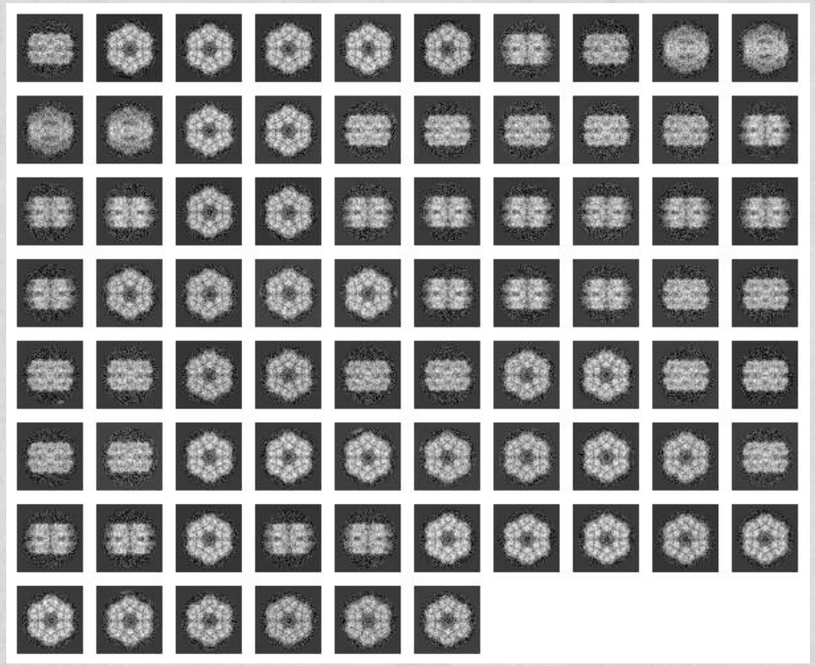
- nella procedura di ottimizzazione di un dato algoritmo di classificazione e' spesso necessario utilizzare procedure di minimizzazione/massimizzazione:
  - ex. l'ottimizzazione consiste nel minimizzare la "distanza" della risposta del classificatore dal valore 1 per eventi segnale e 0 per eventi fondo
  - ex. l'ottimizzazione consiste nel trovare il set di tagli che massimizzano la reiezione del fondo ad un'efficienza fissata per il segnale
  - ....
- L'algoritmo numerico di minimizzazione piu' utilizzato in HEP è l'algoritmo MINUIT, che permette di trovare il minimo di una funzione multi-parametrica e di studiarne la forma intorno al valore minimo
- MINUIT e' uno strumento potente ma tende a diventare instabile al crecere di *n* (numero di variabili di input) a statistica fissata (diradazione della popolazione nello spazio delle variabili di input → minimi locali vicini)
- Per ovviare a tale problema si utilizzano spesso tecniche approssimate alternative:

ALGORITMI DI OTTIMIZZAZIONE

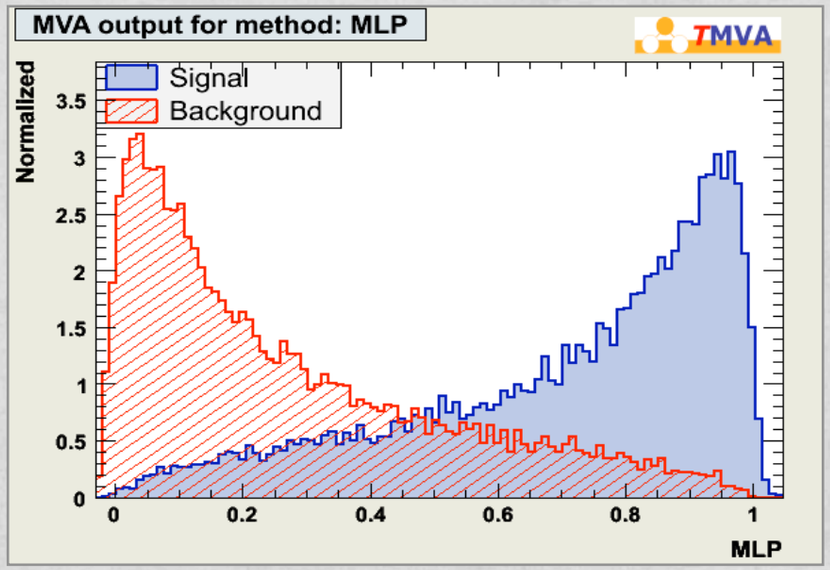
- Campionamento Monte Carlo: campionamento random dell'intero spazio delle variabili di input (priors: uniformi, oppure gaussiani)
  - converge e non e' distorto (trova sempre il minimo giusto), inefficiente (specie al crescere di n)
- Algoritmi genetici: trova soluzioni approssimate a problemi di ottimizzazione, modellando il problema con un set (popolazione) di rappresentazioni astratte (genomi) delle possibili soluzioni, ed applicando schemi evolutivi tipici delle teorie genetiche: ereditarietà, mutazioni, crossover
  - solo le popolazioni piu' forti sopravvivono, algoritmi molto veloci ed efficienti
- Simulated Annealling (ricottura): in metallurgia e' un trattamento a base di calore di un materiale, con il quale si modificano le proprieta' del materiale stesso (durezza, tenacita', ...)
  - l'algoritmo sfrutta l'analogia tra il modo in cui un metallo si raffredda ottenendo un minimo energetico una volta raggiunta la cristallizzazione, ed il problema della ricerca di un minimo assoluto in un sistema generico
- l'idea e' quella di provare variazioni random della soluzione corrente, accettando come nuove soluzioni non solo variazioni che vanno nella direzione giusta (verso il minimo), ma anche una eventuale variazione in direzione sbagliata, con una probabilita' che diminusice mano a mano che il calcolo procede (i.e. la temperatura decresce) → inizialmente variazioni random in tutte le direzioni, alla fine solo verso il minimo
  - vantaggio principale rispetto ad altri metodi: abile nell'evitare di essere intrappolato in minimi locali

CLASSIFICATORI MULTIVARIATI

- L'idea di base di un classificatore multivariato e' quella di condensare un set multidimensionale e correlato di variabili di input in un unica grandezza scalare (output del classificatore):
  - Classificatore multivariato: algoritmo di regressione $R^n \rightarrow R$



$y(H_0) \rightarrow 0$, $y(H_1) \rightarrow 1$



- I classificatori multivariati vengono generalmente divisi in due classi distinte:
  - classificatori lineari: ignorano possibii correlazioni tra variabili (cut-based, PDE) , o sono in grado di gestire al massimo corr. lineari (Fisher, H-Matrix): più veloci, più robusti, meno potenti
  - classificatori non-lineari (mLR, NN, BDT, ...): più potenti, meno veloci, meno robusti

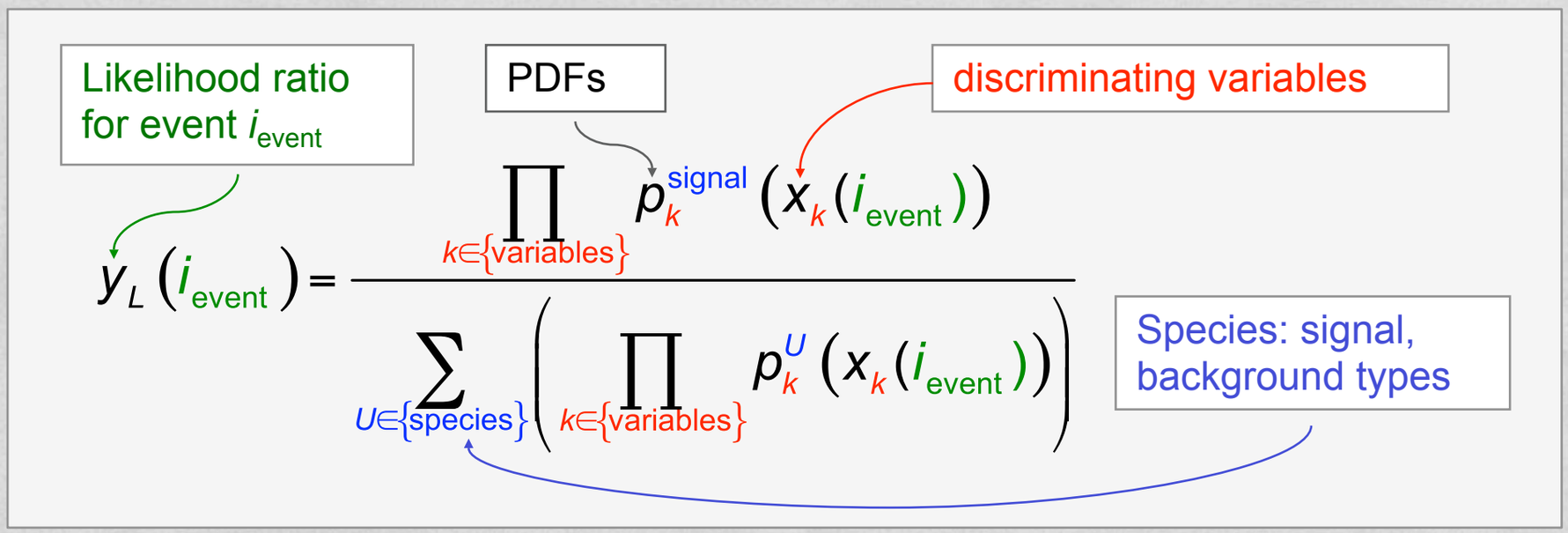
CLASSIFICATORI LINEARI

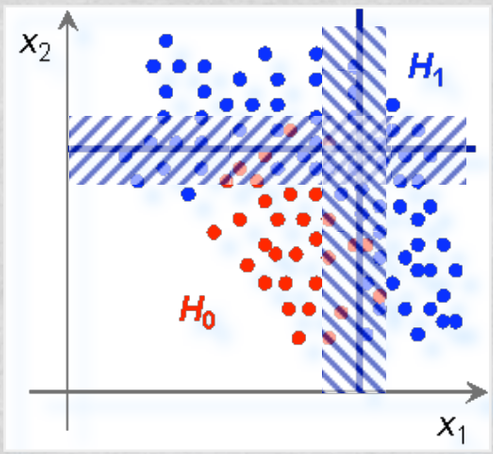
PROJECTIVE LIKELIHOOD ESTIMATOR (PDE)

Idea di base:

- stima della pdf per ogni singola variabile: p_k
- $p_{tot} = \prod p_k$ ← ignora le correlazioni
- combina $p_{tot}(S)$ e $p_{tot}(B)$ in un discriminante tipo Likelihood-Ratio

comportamento ottimale in caso di correlazione zero o lineare (decorrelazione)





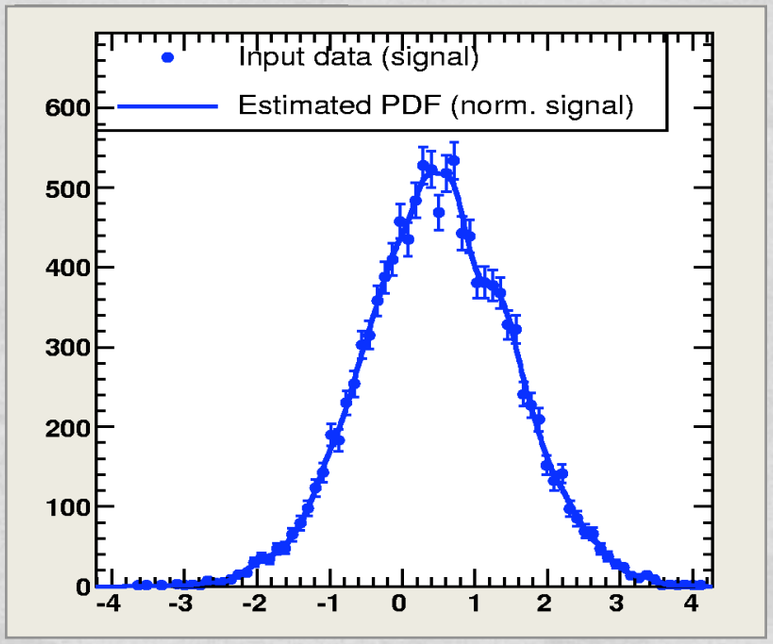
PDE introduces fuzzy logic

PROJECTIVE LIKELIHOOD ESTIMATOR (PDE)

problema principale: come stimare le pdf

- conteggio degli eventi: i.e. histogramma normalizzato usato come una funzione
  - semplice da implementare, non distorto, sub-ottimale
- fit parametrico: necessita di un modello per ogni pdf, difficile da implementare
- stima non parametrica: potente e facile da implementare, puo' creare artefatti (distorto)

Interpolazione basata su spline: binnato



Kernel method:

p(X): pdf da stimare, $x=(x_1,...,x_N)$: insieme di osservazioni sperimentali di p

distribuzione cumulativa di p:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(y)dy$$

una stima della probabilità F è ottenibile contando il numero di punti $x_i : x_i \leq x$:

$$F(x) \sim \hat{F}(x) \doteq \frac{1}{N}\#\{x_i | x_i \leq x\}$$

poichè p(x) = F'(x) $\Rightarrow \exists\ h>0 \in R$:

$$p(x) \sim \frac{1}{2h}(\hat{F}(x+h) - \hat{F}(x-h)) = \frac{1}{2Nh}\#\{x_i | -h < x_i - x \leq h\} \doteq \hat{p}(x)$$

KERNEL METHOD

sia ora per esempio K la funzione di distribuzione uniforme nell'intervallo x∈[0,1]:

$$K(x) = \begin{cases} \frac{1}{2} & -1 \leq x < 1 \\ 0 & altrimenti \end{cases}$$

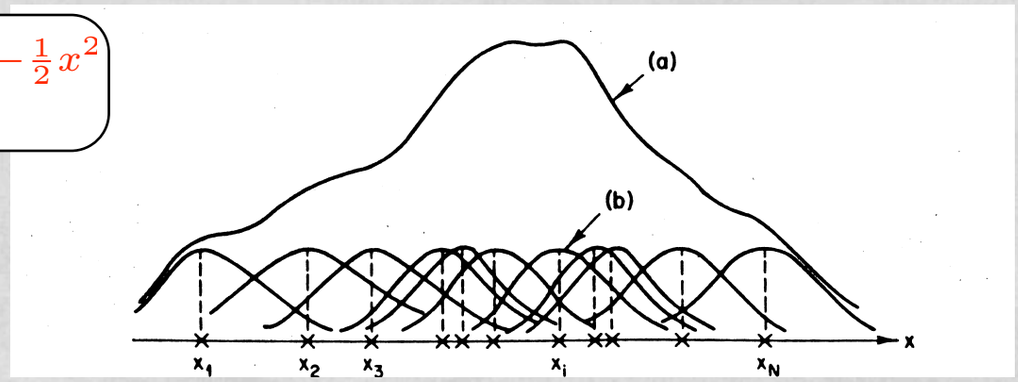
possiamo riscrivere la stima di p(x) come:

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

stima basata sul kernel K della pdf p(x)
con parametro di smoothing h

Nel caso generale K puo' essere una qualunque pdf, la cui scelta dipende dalle caratteristiche del problema che si vuole risolvere: ex. kernel gaussiano

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$



l'ottimizzazione del classificatore consiste nella scelta del parametro di smoothing h ottimale rispetto alla separazione S/B:
NOTA:
h troppo piccolo → "features" inesistenti,
h troppo grande → si perdono dettagli (i.e. potere) della distribuzione

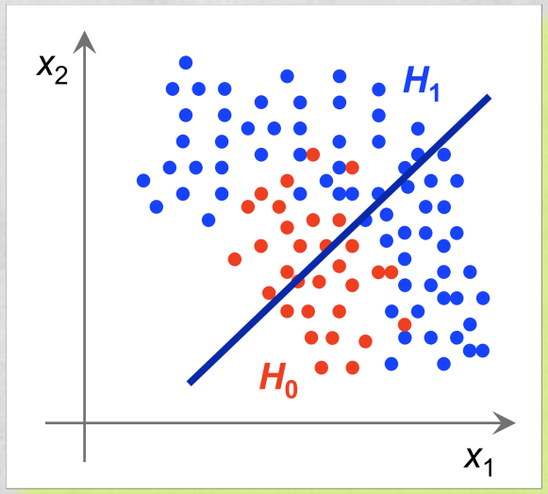
soluzione: smoothing adattivo:
$h = h(x) = h^0/\sqrt{p(x_i)}$ → gaussiane piu' larghe sulle code della pdf

DISCRIMINANTI FISHER

- Un classificatore ben noto, semplice e robusto
- viene trovata una direzione nell'iperspazio delle variabili di input, tale che la proiezione degli eventi su tale direzione massimizzi la "distanza" degli eventi di segnale da quelli di fondo

"Fisher coefficients"

$$y_{\mathrm{Fi}}(i_{\mathrm{event}}) = F_0 + \sum_{k \in \{\mathrm{variables}\}} x_k(i_{\mathrm{event}}) \times F_k$$



- nella procedura di ottimizzazione viene massimizzata la distanza tra le medie delle due classi e minimizzata contemporaneamente la varianza all'interno di ciascuna classe:

$$J(w) = \frac{|m_1 - m_2|^2}{\sigma_1^2 + \sigma_2^2}$$

- se una variabile ha stesso valore medio per segnale e background → nessuna separazione
  - spesso conviene usare |var|
  - prestazioni ottimali per variabili gaussiane con correlazioni lineari

CLASSIFICATORI NON-LINEARI

PDE MULTIDIMENSIONALI

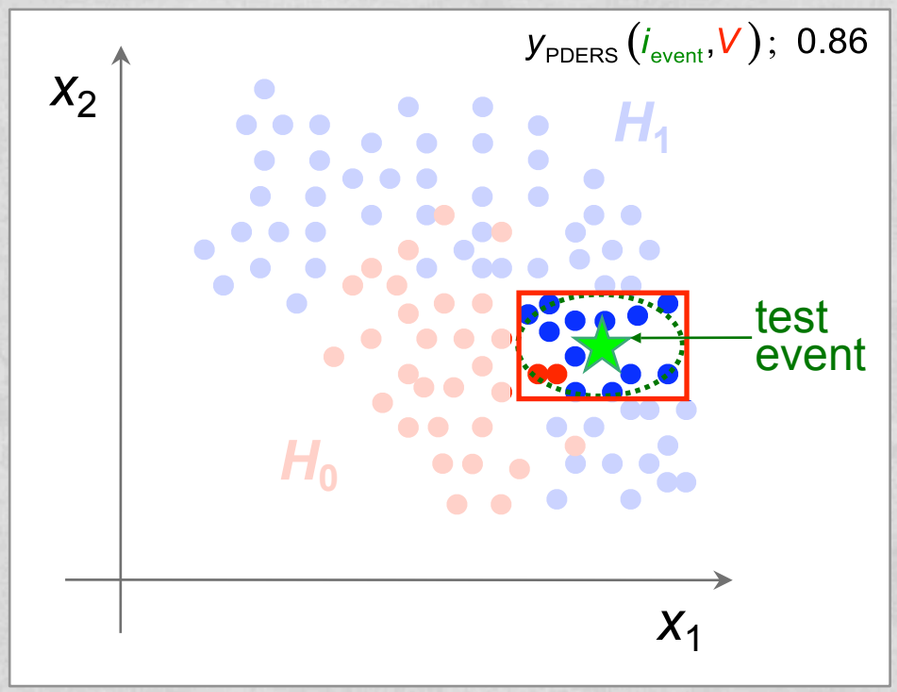
- Multidimensional kernel: generalizzazione del caso 1-D

$$\hat{p}(\underline{x}) = \frac{1}{N h_1 \cdots h_d} \sum_{i=1}^{N} \left\{ \prod_{j=1}^{d} K\left(\frac{x_j - x_{ji}}{h_j} \right) \right\}$$

- PDE-RS (Range-Search):
  - utilizza una singola pdf per segnale e fondo che "spanna" n variabili
  - conteggia il numero di eventi di segnale e fondo che si trovano in "vicinanza" dell'evento di test
  - un iper-volume fissato o adattivo definisce la "vicinanza"
  - lo stimatore del segnale e' dato da:

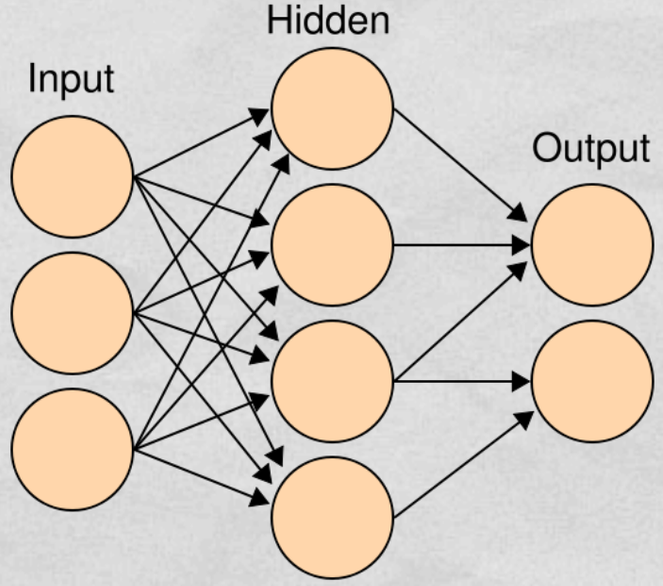
PDE-RS ratio for event i_{event} — chosen volume — #signal events in V — #background events in V

$$y_{PDERS}(i_{event}, V) = \frac{n_S(i_{event}, V)}{n_S(i_{event}, V) + n_B(i_{event}, V)}$$



ARTIFICIAL NEURAL NETWORKS

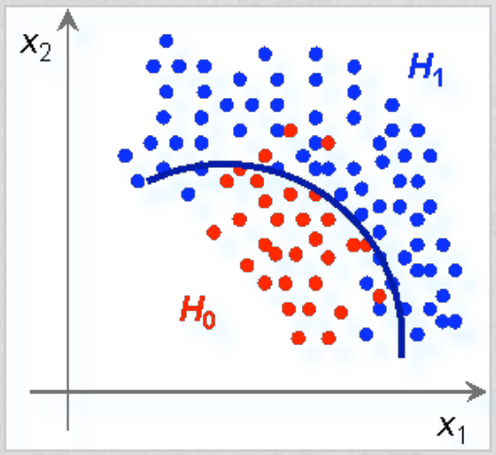
- L'esempio piu' noto di classificatore non-lineare
- Un ANN e' un modello matematico basato sulla similitudine con i network neurali biologici:
  - consiste in un gruppo interconnesso di neuroni artificiali
  - analizza informazioni in input in accordo ad un approccio computazionale di tipo connessionista → azioni eseguite in modo collettivo ed in parallelo da unita' semplici (neuroni)
  - si comporta come un sistema adattivo: modifica la sua struttura basandosi sull'insieme di informazioni che fluiscono attraverso la rete durante la fase di apprendimento



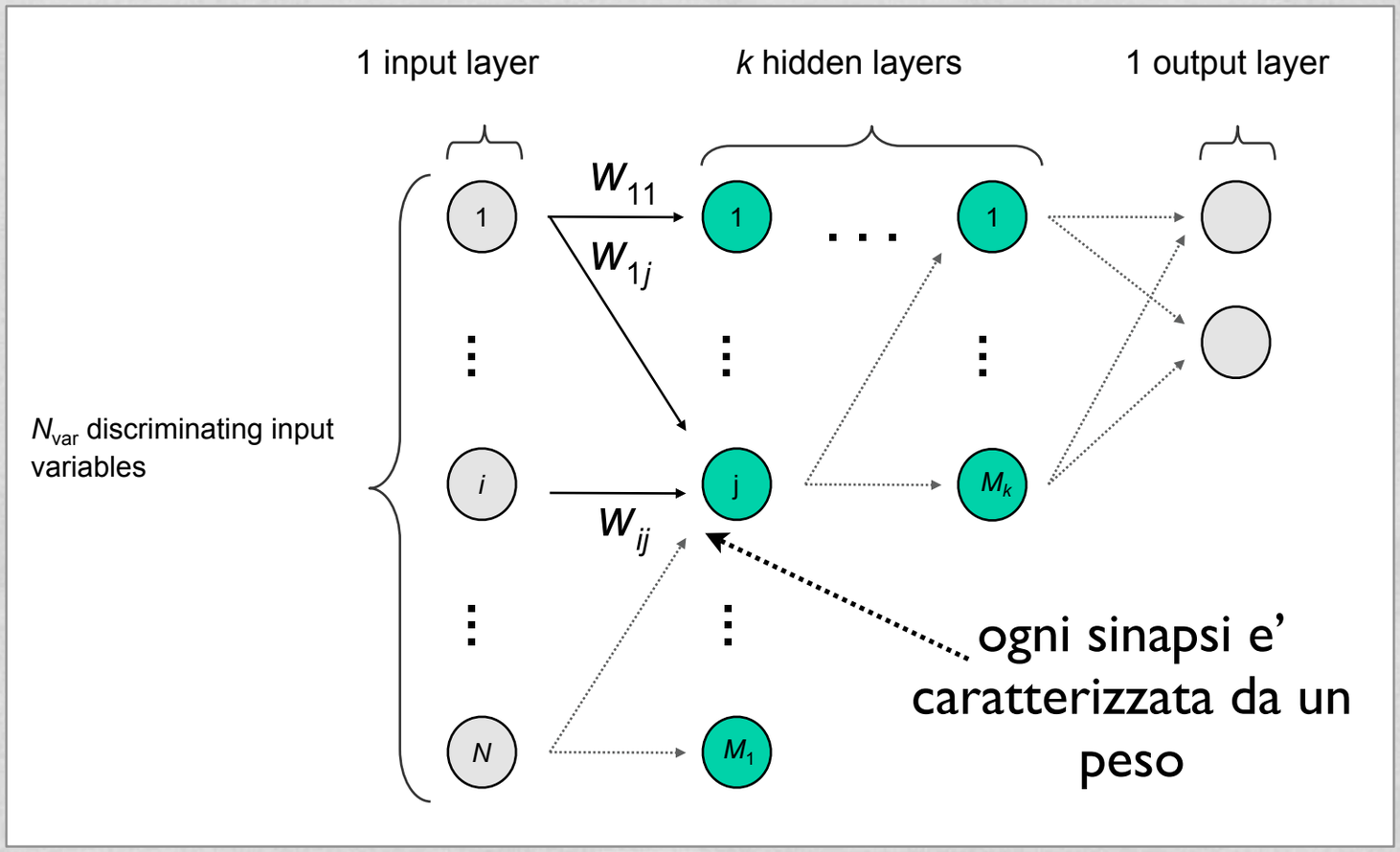
- la risposta non lineare viene ottenuta attraverso l'attivazione di nodi di output tramite pesi non lineari

FEED-FORWARD ANN

- In HEP tutti gli ANN hanno una struttura di tipo Feed-Forward multilayer perceptron:
  - neuroni organizzati in layers: input, hidden-1, ... , hidden-K, output
  - sono possibili solo connessioni da un dato layer a quello immediatamente successivo



Teorema di Weierstrass:
e' possibile approssimare qualunque funzione continua (con precisione arbitraria) , usando un singolo layer nascosto ed un numero infinito di neuroni



Feed-forward Multilayer Perceptron

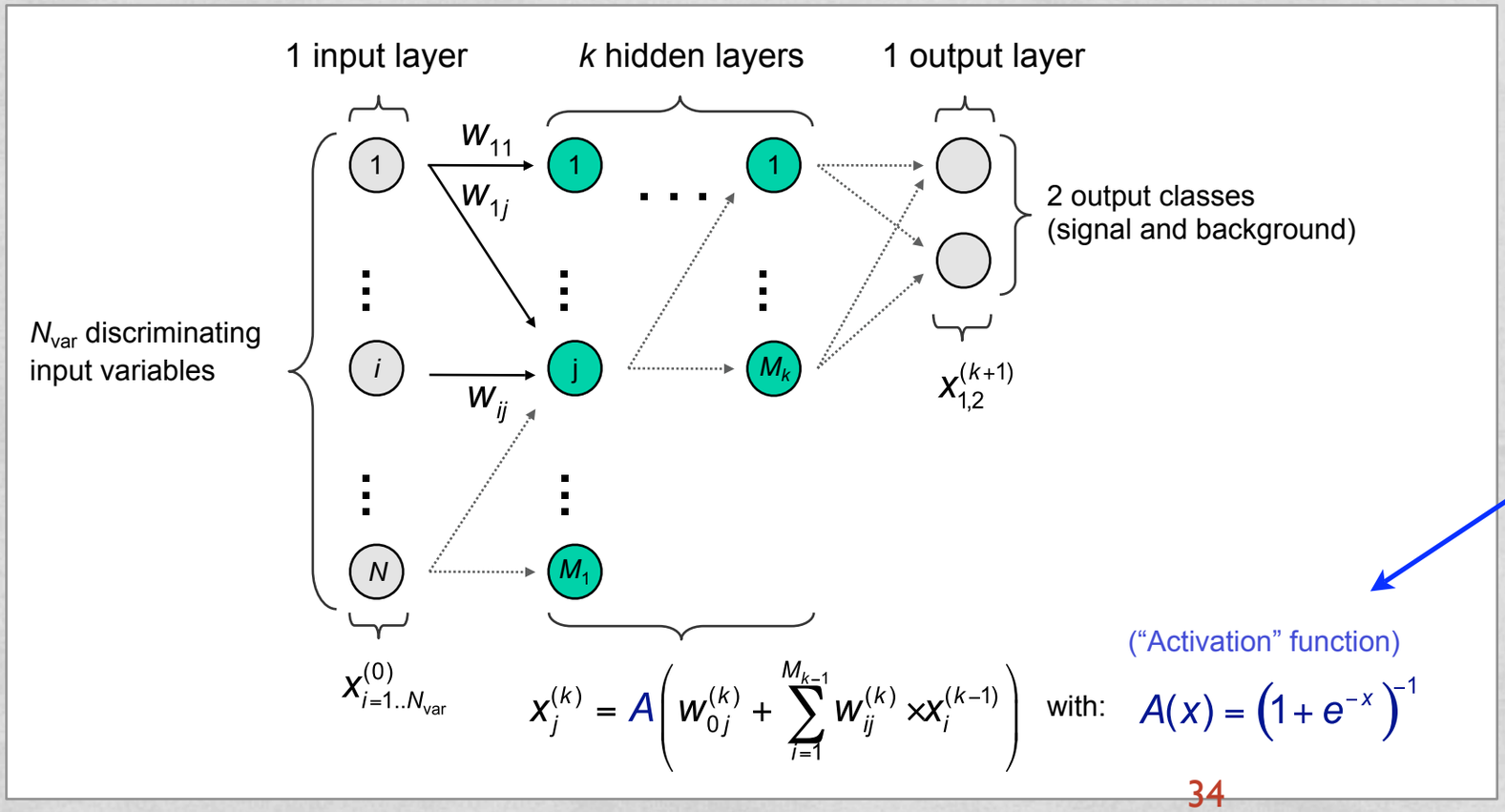
FUNZIONE DI RISPOSTA

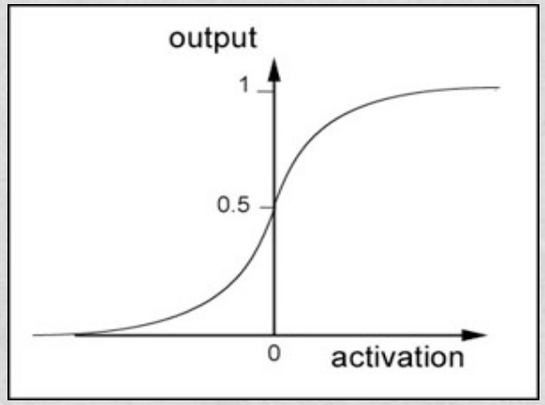
- comportamento di una rete neurale determinato da: struttura topologica dei neuroni, pesi asociati alle connessioni tra neuroni, risposta dei neuroni ai dati di input
- Funzione di risposta ρ:
  - mappa l'input al neurone $x^{(k-1)}_1,...,x^{(k-1)}_n$ nell'output del neurone $x^{(k)}_j$
  - viene normalmente separata in una funzione sinaptica $k:R^n \rightarrow R$ ed una funzione di attivazione neurale $A:R \rightarrow R$: $\rho = k \bullet A$

$$A : x \rightarrow \begin{cases} x & Lineare \\ \frac{1}{1+e^{-kx}} & Sigmoide \\ \frac{e^x - e^{-x}}{e^x + e^{-x}} & Tanh \\ e^{-x^2/2)} & Radiale \end{cases}$$

FUNZIONE DI RISPOSTA

- comportamento di una rete neurale determinato da: struttura topologica dei neuroni, pesi asociati alle connessioni tra neuroni, risposta dei neuroni ai dati di input
- Funzione di risposta ρ:
  - mappa l'input al neurone $x^{(k-1)}_1,...,x^{(k-1)}_n$ nell'output del neurone $x^{(k)}_j$
  - viene normalmente separata in una funzione sinaptica $k:R^n \rightarrow R$ ed una funzione di attivazione neurale $A:R \rightarrow R$: $\rho = k \bullet A$





$$A : x \rightarrow \begin{cases} x & Lineare \\ \frac{1}{1+e^{-kx}} & Sigmoide \\ \frac{e^x - e^{-x}}{e^x + e^{-x}} & Tanh \\ e^{-x^2/2)} & Radiale \end{cases}$$

ADDESTRAMENTO DI UN NN

- L'operazione di aggiustamento dei pesi di ogni singola sinapsi e' chiamata training della rete neurale
- Durante la fase di training i pesi vengono aggiustati in modo da ottimizzare la risposta del classificatore (i.e. massimizzare la separazione Segnale-Fondo per esempio)
- Tecnica piu utilizzata per il training: Back-propagation

Output per un ANN con:
- singolo layer nascosto con A: tanh
- output layer con A: lineare

n_h: numero neuroni layer nascosto

n_{var}: numero neuroni input layer

$$y_{ANN} = \sum_{j=1}^{n_h} x_j^{(2)} w_{j1}^{(2)} = \sum_{j=1}^{n_h} \tanh\left(\sum_{i=1}^{n_{var}} x_i w_{ij}^{(1)}\right) w_{j1}^{(2)}$$

peso tra j-esimo neurone layer nascosto e neurone di output

peso tra i-esimo neurone layer input e j-esimo neurone layer nascosto

TRAINING (2)

- durante il processo di addestramento del NN vengono forniti al network N eventi: $\boldsymbol{x}_a$ (a=1,..,N) appartenenti alle classi di segnale e fondo
- per ogni evento di training viene calcolato $y_{ANN}(a)$ e confrontato con l'output aspettato: $Y_a \in \{0,1\}$ (0 per eventi fondo, 1 per eventi segnale)
- Viene definita una misura della distanza tra $y_{ANN}(a)$ e Y_a come:

$$\Delta(x_1,...,x_N|\mathbf{w}) = \sum_{\mathbf{a=1}}^{\mathbf{N}} \boldsymbol{\Delta}_{\mathbf{a}}(\mathbf{x_a}|\mathbf{w}) = \sum_{\mathbf{a=1}}^{\mathbf{N}} \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{y_{ANN}(a)} - \mathbf{Y_a})^{\mathbf{2}}$$

- il set di pesi w ottimizzato e' quello che minimizza la distanza Δ
  - minimizzazione ottenuta con metodi steepest descent/gradient descent

$$\mathbf{w}^{(\rho+\mathbf{1})} = \mathbf{w}^{(\rho)} - \eta \nabla_{\mathbf{w}} \boldsymbol{\Delta}$$

dato un set random di pesi $w^{(\rho)}$, i pesi sono aggiornati muovendosi di una piccola distanza in direzione $\nabla_w \Delta$

ALBERI DI DECISIONI BINARIE (DECISION TREES)

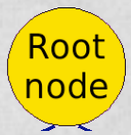
- IDEA: applicazione sequenziale di decisioni di tipo binario ⇒ suddivide i dati in nodi. I nodi terminali (foglie) possano essere utilizzati per classificare l'evento
  - ☺ semplice interpretazione (rappresentazione grafica in 2D)
  - ☺ non e' sensibile a variabili deboli, cioè che non hanno potere di separazione
  - ☹ instabile: piccole variazioni del sample di training hanno grandi effetto sulla risposta
  - ☹ soffre di overtraining: utilizzo di tecniche di potatura per minimizzare l'effetto

Costruzione di un decision tree:

- start: root node
- split del campione di training con un taglio sulla migliore variabile associata al nodo
- criterio di divisione: Gini-index: $p \times (1-p)$
- splitting continua fino a quando si raggiunge un numero minimo di eventi o quando $p=p_{max}$
- ogni foglia viene classificata in accordo alla maggioranza degli eventi che vi cadono, oppure con pesi $\propto$ al numero di eventi S e B che cadono nella foglia

ALBERI DI DECISIONI BINARIE (DECISION TREES)

- IDEA: applicazione sequenziale di decisioni di tipo binario ⇒ suddivide i dati in nodi. I nodi terminali (foglie) possano essere utilizzati per classificare l'evento
  - ☺ semplice interpretazione (rappresentazione grafica in 2D)
  - ☺ non e' sensibile a variabili deboli, cioè che non hanno potere di separazione
  - ☹ instabile: piccole variazioni del sample di training hanno grandi effetto sulla risposta
  - ☹ soffre di overtraining: utilizzo di tecniche di potatura per minimizzare l'effetto



Costruzione di un decision tree:

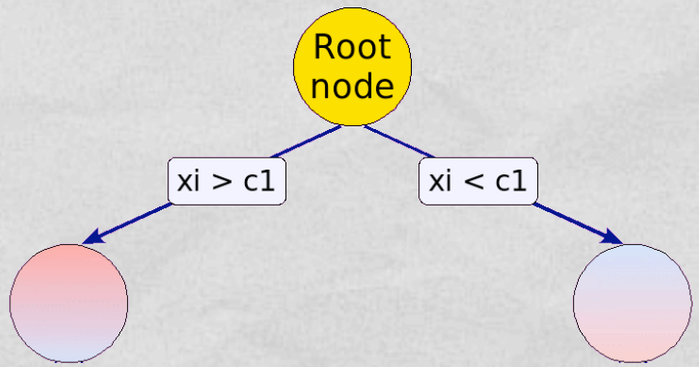
- start: root node
- split del campione di training con un taglio sulla migliore variabile associata al nodo
- criterio di divisione: Gini-index: $p \times (1-p)$
- splitting continua fino a quando si raggiunge un numero minimo di eventi o quando $p=p_{max}$
- ogni foglia viene classificata in accordo alla maggioranza degli eventi che vi cadono, oppure con pesi $\propto$ al numero di eventi S e B che cadono nella foglia

ALBERI DI DECISIONI BINARIE (DECISION TREES)

- IDEA: applicazione sequenziale di decisioni di tipo binario ⇒ suddivide i dati in nodi. I nodi terminali (foglie) possano essere utilizzati per classificare l'evento
  - ☺ semplice interpretazione (rappresentazione grafica in 2D)
  - ☺ non e' sensibile a variabili deboli, cioè che non hanno potere di separazione
  - ☹ instabile: piccole variazioni del sample di training hanno grandi effetto sulla risposta
  - ☹ soffre di overtraining: utilizzo di tecniche di potatura per minimizzare l'effetto

Costruzione di un decision tree:

- start: root node
- split del campione di training con un taglio sulla migliore variabile associata al nodo
- criterio di divisione: Gini-index: $p \times (1-p)$
- splitting continua fino a quando si raggiunge un numero minimo di eventi o quando $p=p_{max}$
- ogni foglia viene classificata in accordo alla maggioranza degli eventi che vi cadono, oppure con pesi $\propto$ al numero di eventi S e B che cadono nella foglia

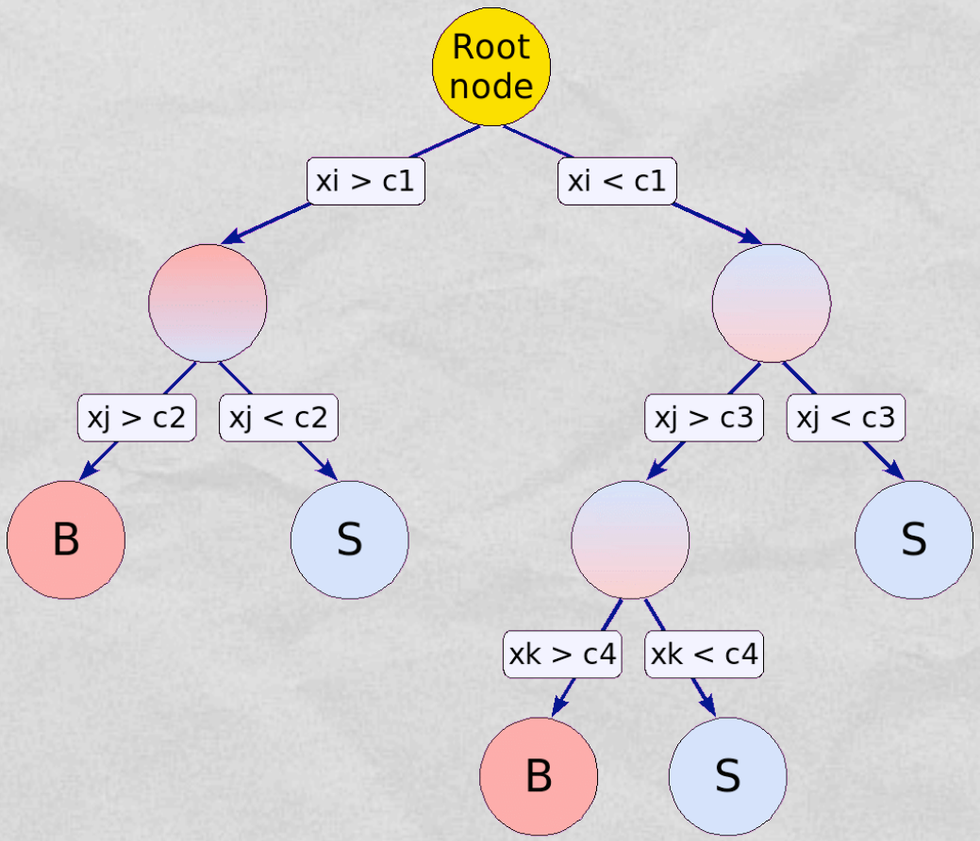


ALBERI DI DECISIONI BINARIE (DECISION TREES)

- IDEA: applicazione sequenziale di decisioni di tipo binario ⇒ suddivide i dati in nodi. I nodi terminali (foglie) possano essere utilizzati per classificare l'evento
  - ☺ semplice interpretazione (rappresentazione grafica in 2D)
  - ☺ non e' sensibile a variabili deboli, cioè che non hanno potere di separazione
  - ☹ instabile: piccole variazioni del sample di training hanno grandi effetto sulla risposta
  - ☹ soffre di overtraining: utilizzo di tecniche di potatura per minimizzare l'effetto

Costruzione di un decision tree:

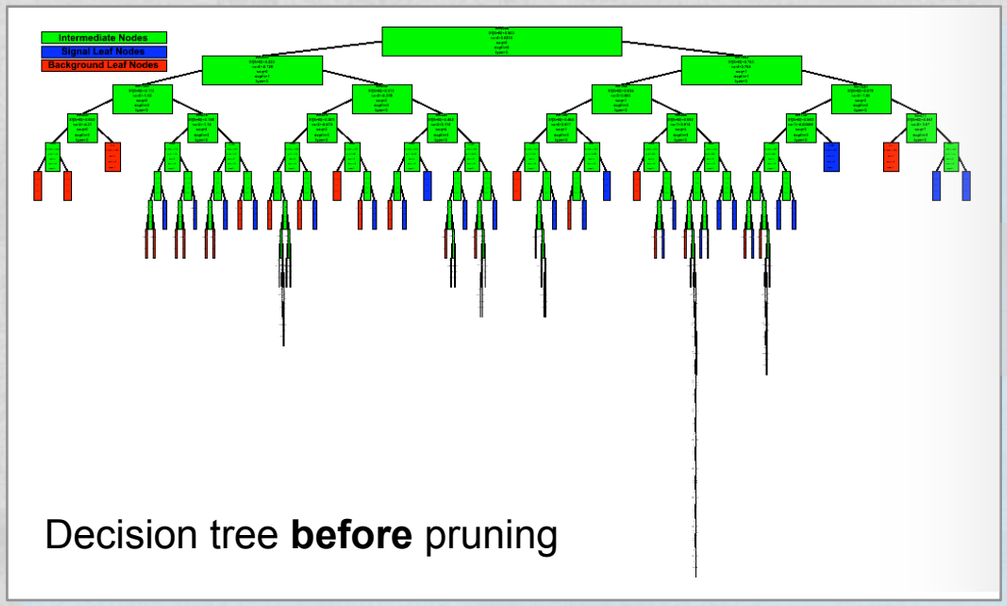
- start: root node
- split del campione di training con un taglio sulla migliore variabile associata al nodo
- criterio di divisione: Gini-index: $p\times(1-p)$
- splitting continua fino a quando si raggiunge un numero minimo di eventi o quando $p=p_{max}$
- ogni foglia viene classificata in accordo alla maggioranza degli eventi che vi cadono, oppure con pesi $\propto$ al numero di eventi S e B che cadono nella foglia

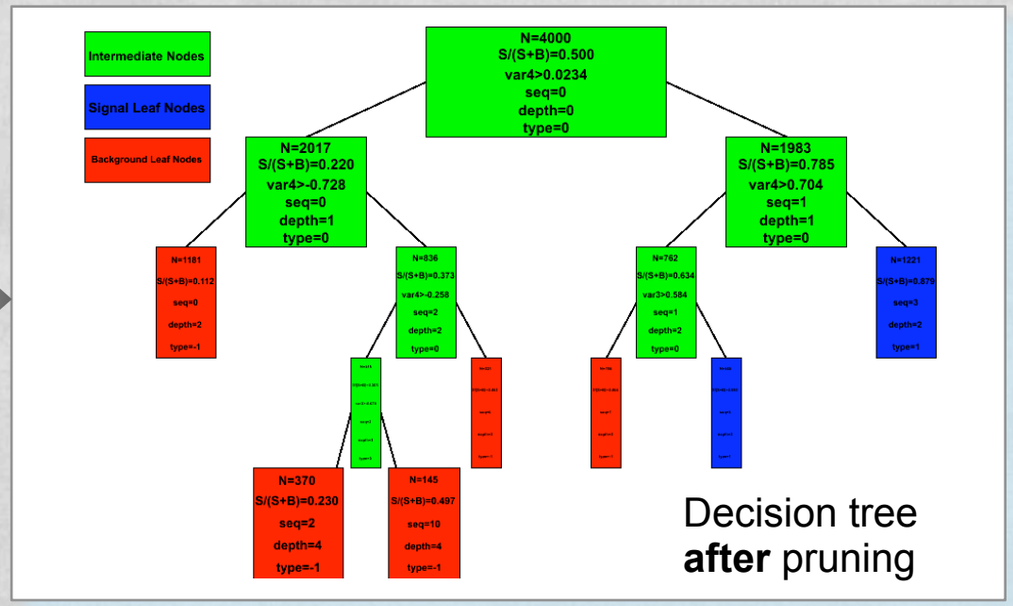


PRUNING

- per minimizzare la sensibilita' dei DT alle fluttuazioni statistiche nei campioni di training e' utile rimuovere (potare) alcuni nodi dopo che l'albero e' stato fatto crescere fino alla sua dimensione massima ⇒ minimizza la probabilita' di overtraining del tree
  - 2 algoritmi di potatura:
    - expected error: vengono tagliati tutti i nodi per i quali la stima dell'errore statistico dei nodi parenti e' minore di quello combinato dei nodi figli (err = $\sqrt{(p(1-p)/N)}$)
    - cost complexity: vengono tagliati i nodi con il piu' piccolo valore di ρ

R≡rate di mis-classificazione $\rho = \frac{R(nodo) - R(sotto\ nodi)}{(sotto\ nodi) - 1}$





BOOSTING: BDT

- procedura per la quale lo stesso classificatore viene addestrato piu' volte, ogni volta usando un campione di eventi di training ripesato (boosted)
- il classificatore finale e' derivato dall'insieme dei classificatori costruiti con la tecnica di boost
  - ADA boost (adaptive boost):
    - gli eventi che vengono classificati in modo erroneo durante la crescita di un tree, ricevono un peso maggiore nella crescita (training) del tree successivo

- Sia $h_i(\mathbf{x}) \in \{0,1\}$ la risposta di un dato tree al vettore di osservabili di input $\mathbf{x}$:

$$y_{BDT}(\mathbf{x}) = \sum_{i \in forest} w_i h_i(\mathbf{x})$$

output del BDT:
piccoli valori di y_{BDT} → fondo
grandi valori di y_{BDT} → segnale

RULE-FIT

Friedman-Popescu

- Evoluzione degli alberi di decisioni binarie:
  - usare un'insieme di regole (rules) per costruire una funzione obiettivo che abbia il massimo potere di separazione tra due classi
  - modello: sequenza lineare di regole, in cui regola ≡ sequenza di tagli

RuleFit classifier

rules (cut sequence → r_m=1 if all cuts satisfied, =0 otherwise)

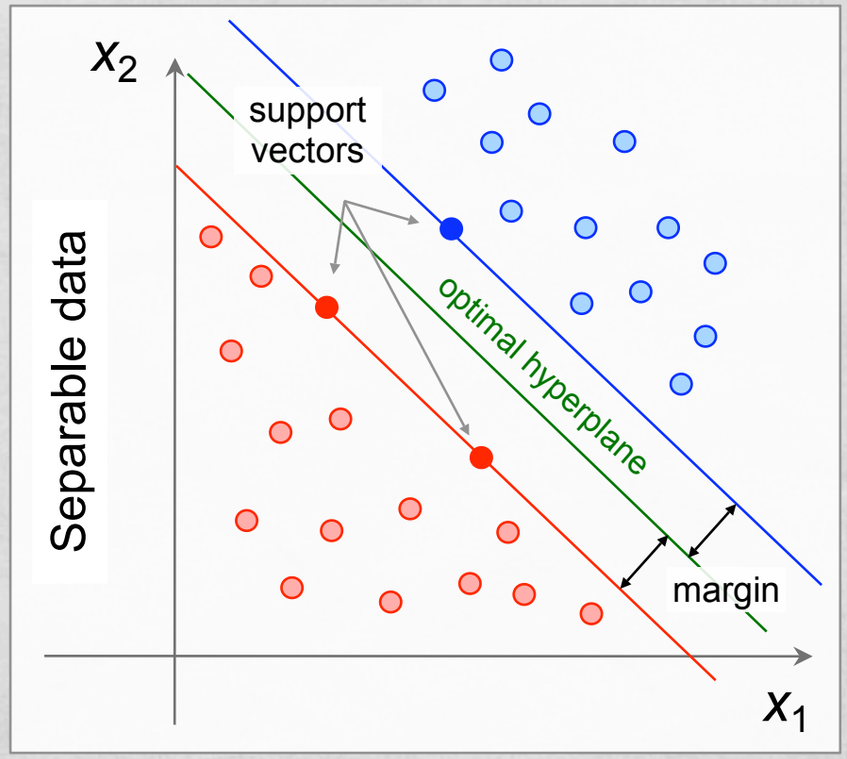
normalised discriminating event variables

$$y_{RF}(\mathbf{x}) = a_0 + \underbrace{\sum_{m=1}^{M_R} a_m r_m(\mathbf{x})}_{\text{Sum of rules}} + \underbrace{\sum_{i=1}^{n_{var}} b_i x_i}_{\text{Linear Fisher term}}$$

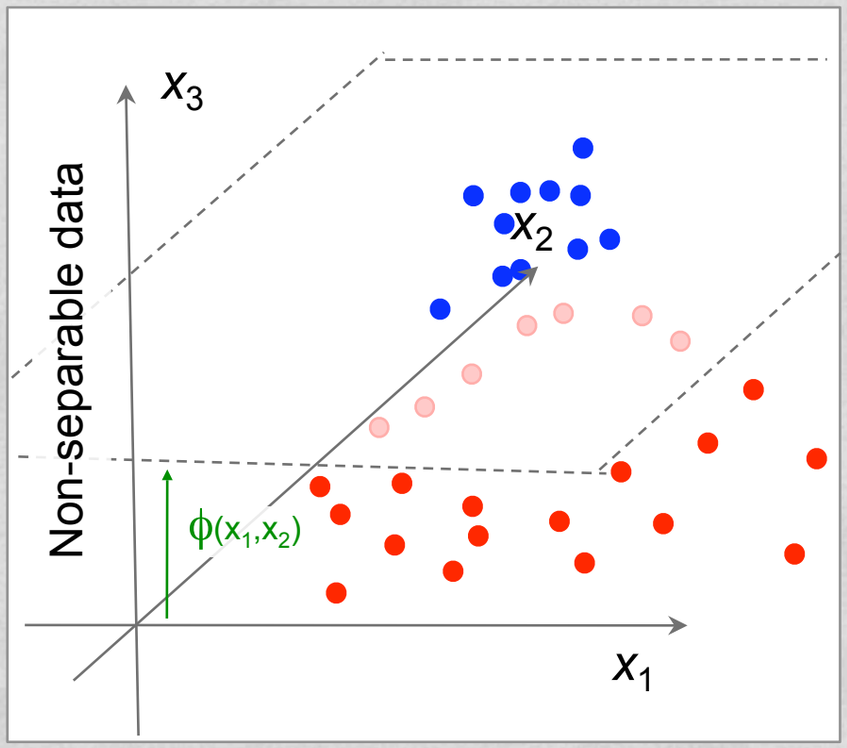
- regole estratte da una foresta di decision trees (creata con tecniche di boosting)
- regole topologicamente simili (i.e. stesse variabili nella sequenza di tagli) eliminate applicando algoritmi di pruning
- i coefficienti a_k e b_k vengono ottimizzati attraverso la minimizzazione del rischio di mis-classificazione rispetto ai caampioni di training

SUPPORT VECTOR MACHINES

- Idea principale: costruire un iperpiano che separi i vettori (eventi) di segnale e fondo, utilizzando solo un subset minimale di tutti i vettori di training (vettori di supporto)



- caso lineare:
  - posizione dell'iperpiano quella che massimizza la distanza (margine) tra esso e i vettori di supporto



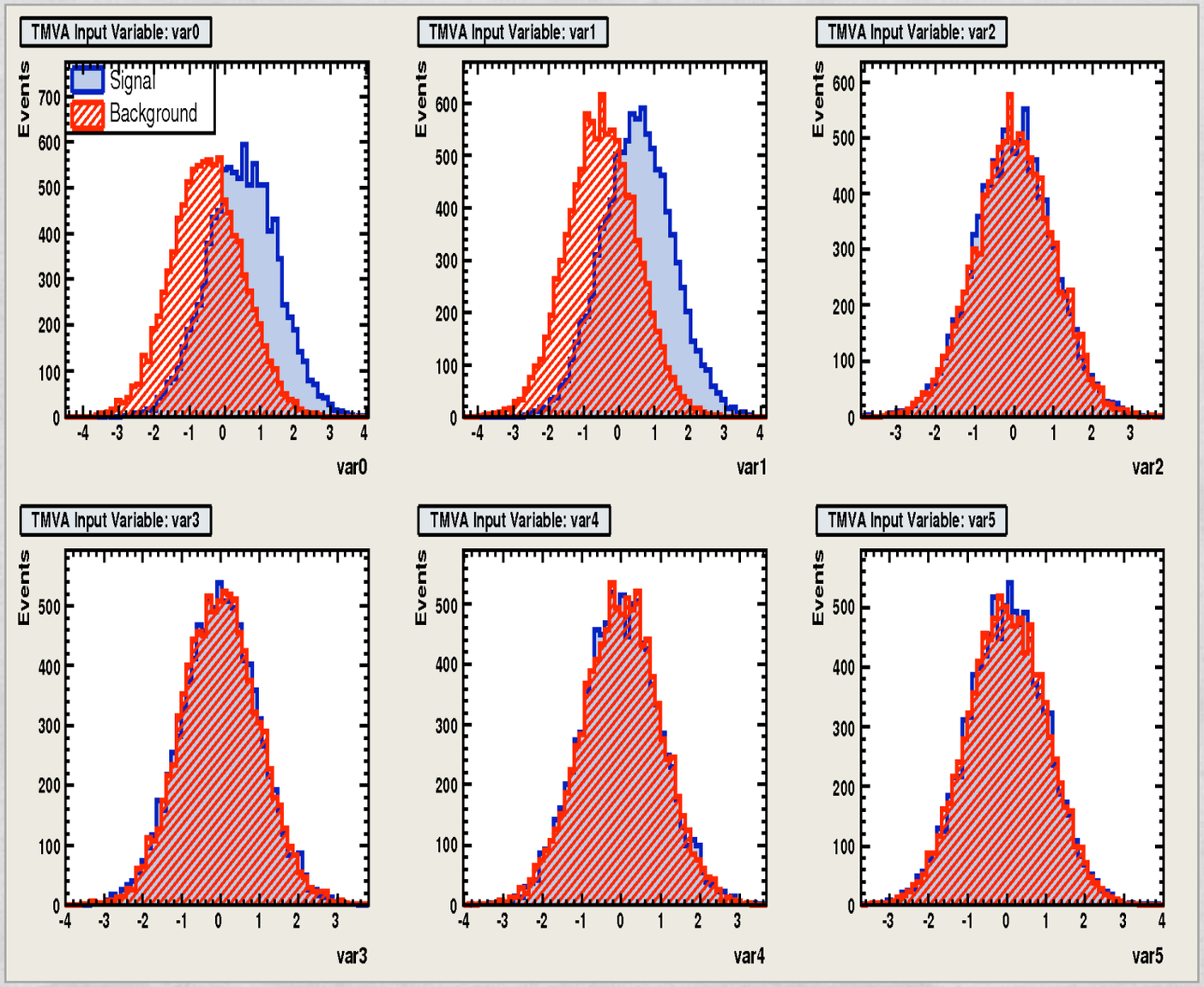
- estensione al caso non lineare:
  - mappatura delle variabili di input in uno spazio di dimensione maggiore nel quale gli eventi di segnale e fondo possono essere separati in modo ottimale con la tecnica del caso lineare

PROPRIETA' DEI CLASSIFICATORI

- Trasparenza: intuitivita' e chiarezza dell'algoritmo e della dipendenza della riposta dall'input
- Prestazioni: i.e. potere di separazione tra le due classi
  - potere di separazione, rarita', $(1-\varepsilon_B)$ VS ε_S
- Velocita': tempo di training, tempo di risposta
- Robustezza: stabilita' delle prestazioni del classificatore rispetto a variazioni nei parametri utilizzati
  - overtraining: compare quando il classificatore ha troppi parametri modificabili e troppo pochi eventi di training
    - falso aumento del potere di separazione
  - variabili deboli: sensibilita' a variabili irrilevanti
- Curse of dimensionality: quanto bene un classificatore risponde all'aumento delle variabili di input (velocita', numero di eventi di training, ...)

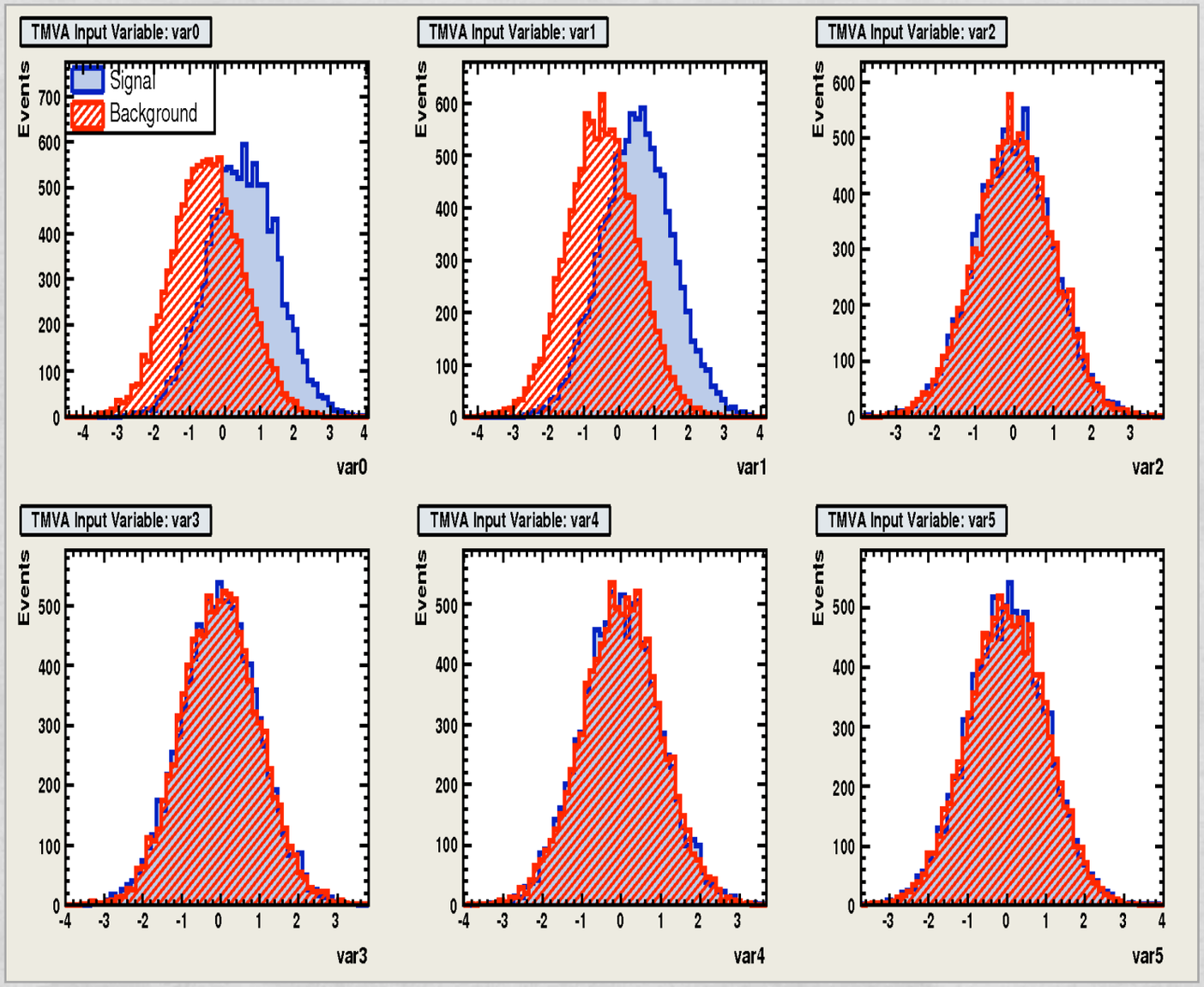
STABILITA' RISPETTO A VARIABILI IRRILEVANTI

Toy con 2 variabili buone e 4 irrilevanti

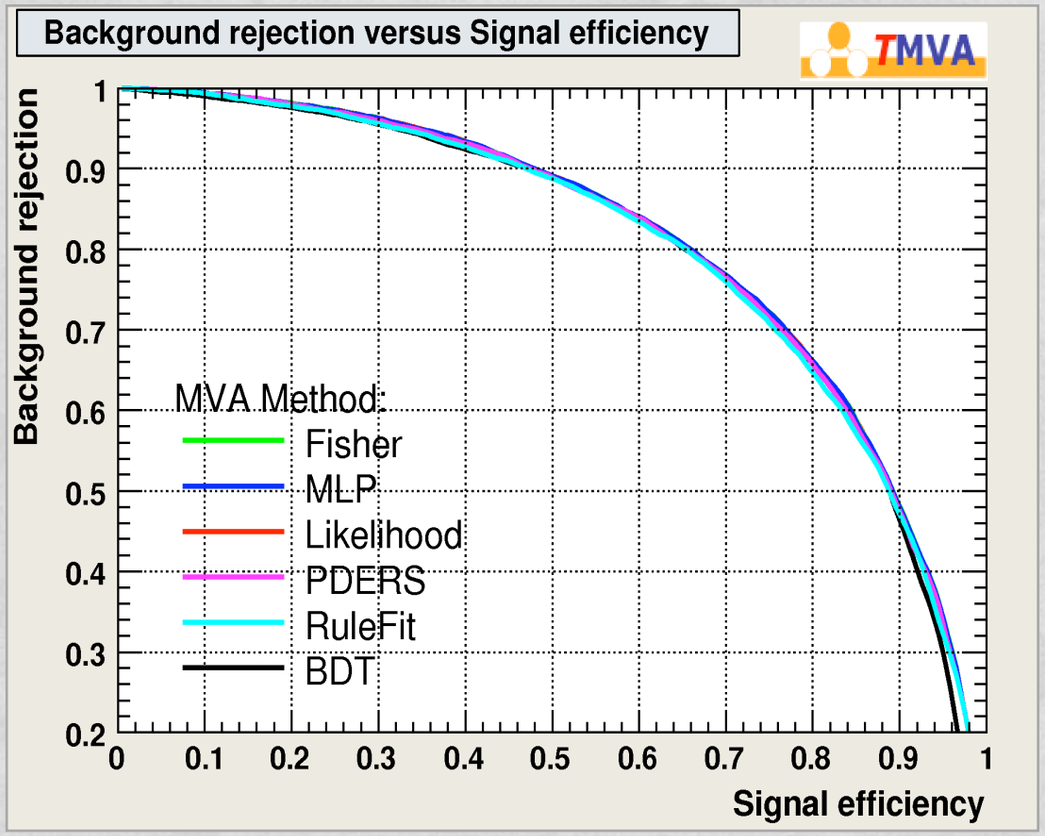


STABILITA' RISPETTO A VARIABILI IRRILEVANTI

Toy con 2 variabili buone e 4 irrilevanti

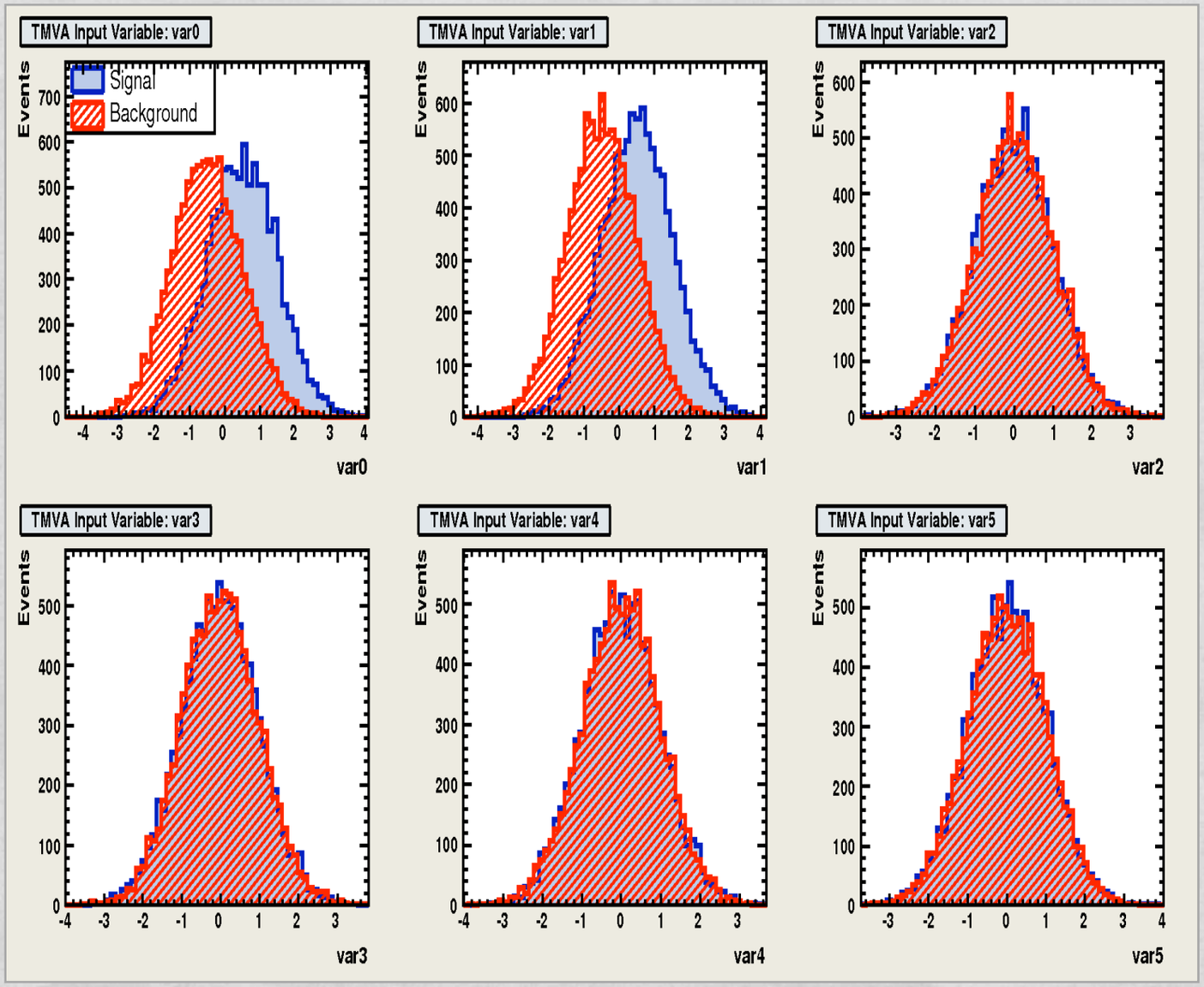


Classificatori addestrati con solo var0 e var1

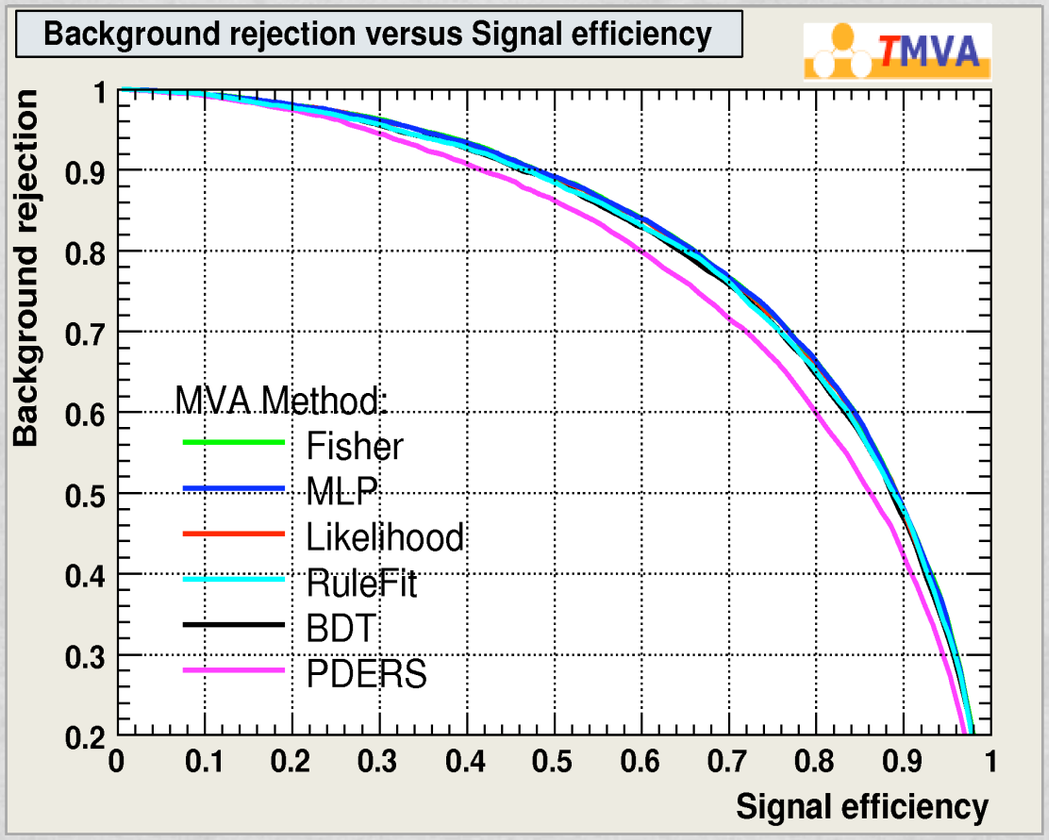


STABILITA' RISPETTO A VARIABILI IRRILEVANTI

Toy con 2 variabili buone e 4 irrilevanti

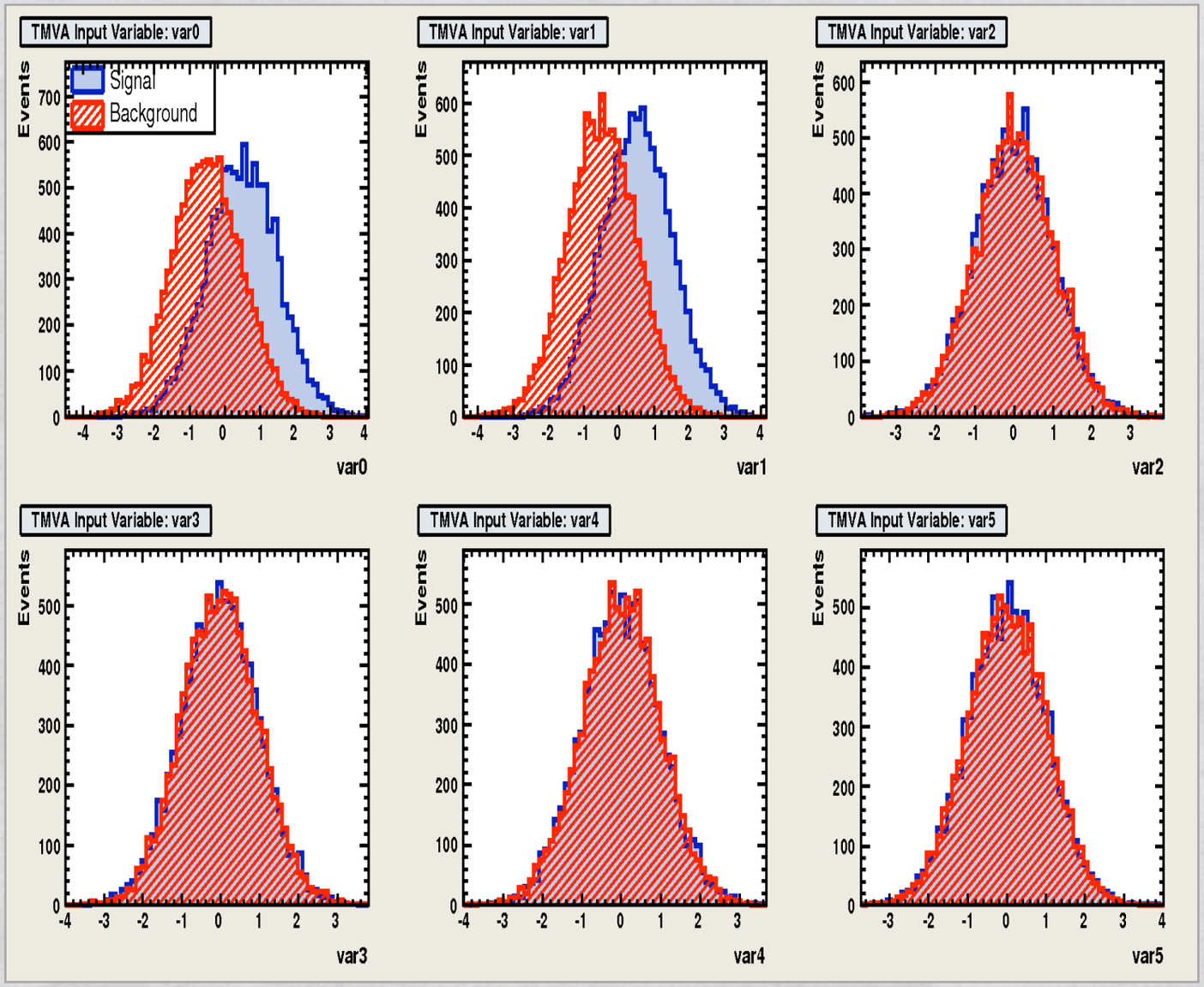


classificatori addestrati con tutte le variabili

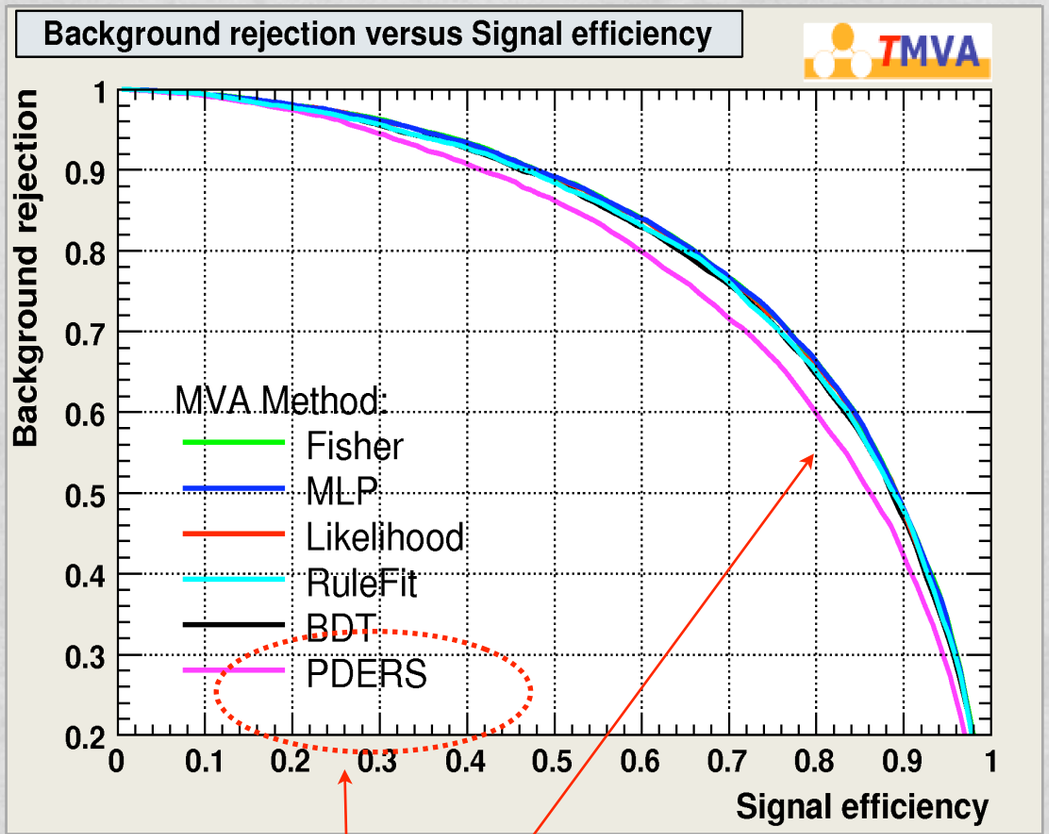


STABILITA' RISPETTO A VARIABILI IRRILEVANTI

Toy con 2 variabili buone e 4 irrilevanti



classificatori addestrati con tutte le variabili



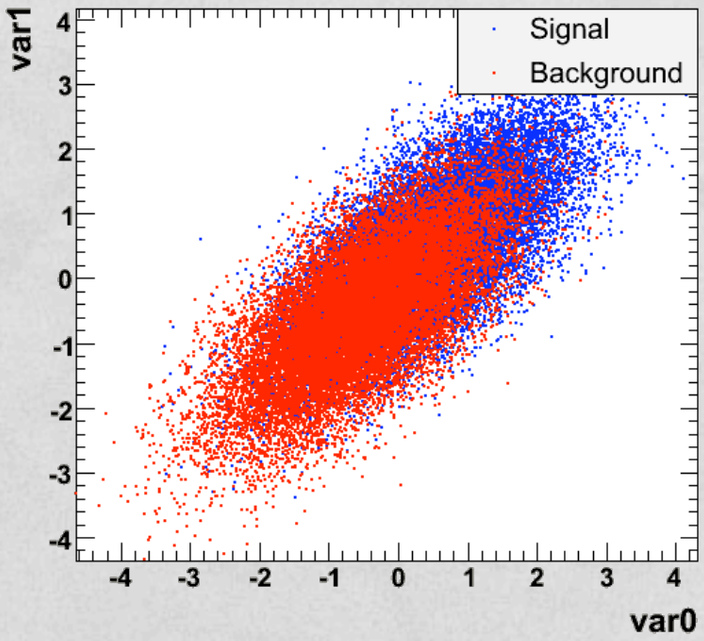
PDE-RS molto sensibile a variabili deboli

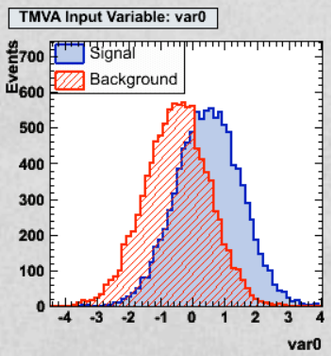
PROPRIETA DEI CLASSIFICATORI MULTIVARIATI

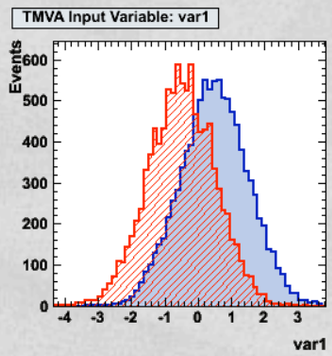
Criteria		Classifiers								
		Cuts	Likeli-hood	PDERS/ k-NN	H-Matrix	Fisher	ANN	BDT	RuleFit	SVM
Perfor-mance	no / linear correlations	😐	🙂	🙂	😐	🙂	🙂	😐	🙂	🙂
	nonlinear correlations	😐	☹	🙂	☹	☹	🙂	🙂	😐	🙂
Speed	Training	☹	🙂	🙂	🙂	🙂	😐	☹	😐	☹
	Response	🙂	🙂	☹/😐	🙂	🙂	🙂	😐	😐	😐
Robust-ness	Overtraining	🙂	😐	😐	🙂	🙂	☹	☹	😐	😐
	Weak input variables	🙂	🙂	☹	🙂	🙂	😐	😐	😐	😐
Curse of dimensionality		☹	🙂	☹	🙂	🙂	😐	🙂	😐	😐
Transparency		🙂	🙂	😐	🙂	🙂	☹	☹	☹	☹

ESEMPIO
CLASS. LINEARI VS NON-LINEARI

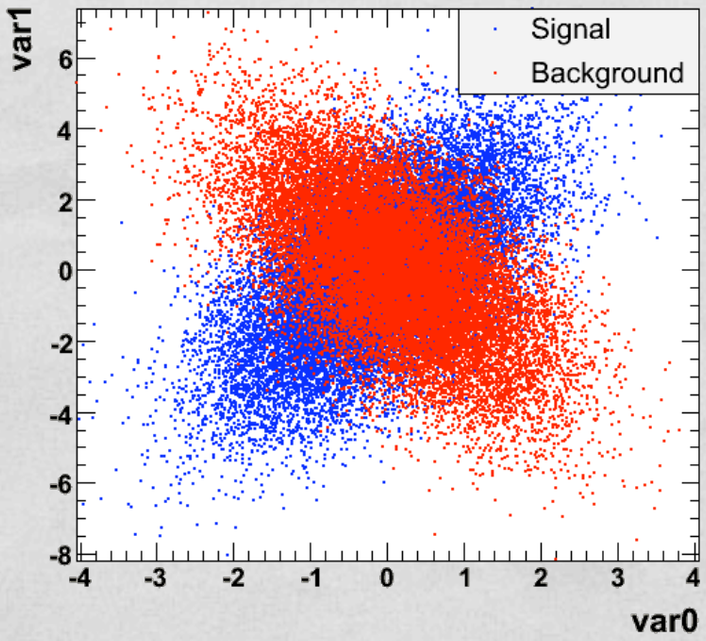
Linear correlations
(same for signal and background)

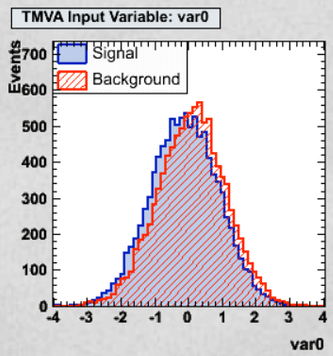


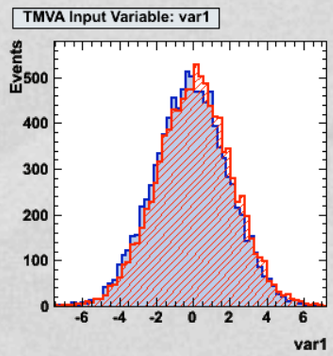




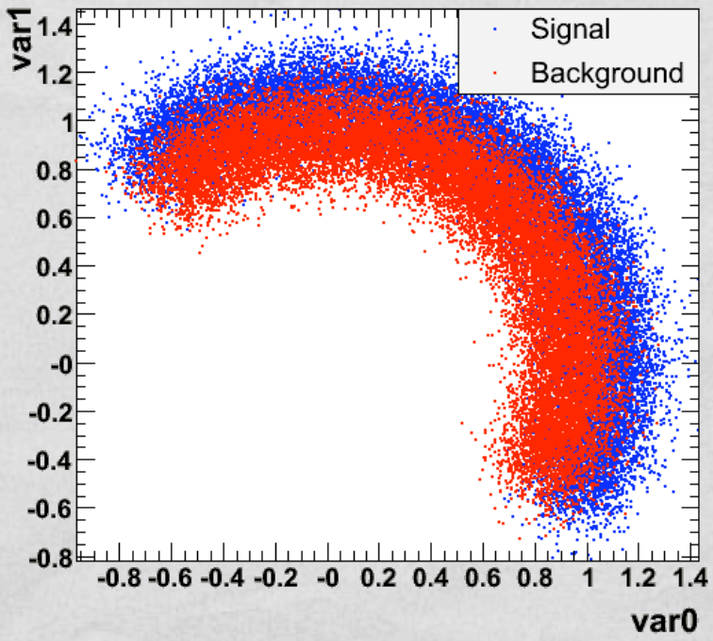
Linear correlations
(opposite for signal and background)

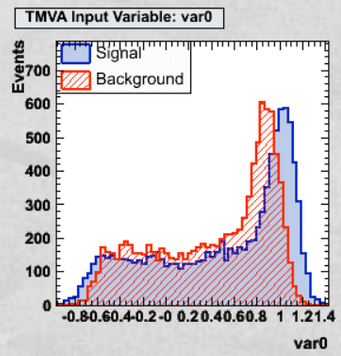


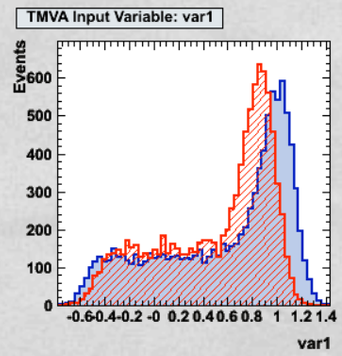




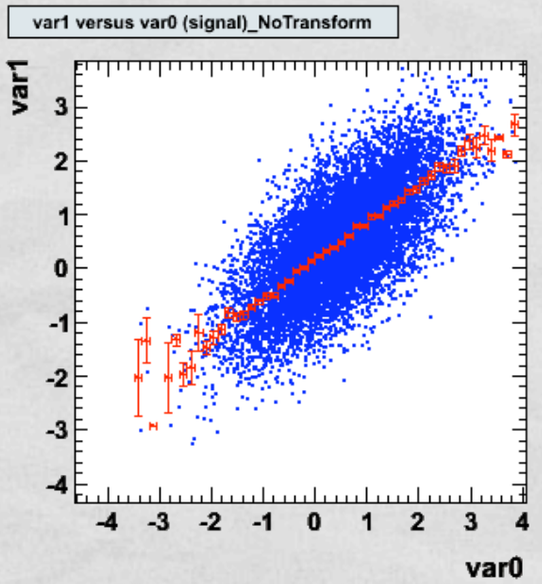
Circular correlations
(same for signal and background)

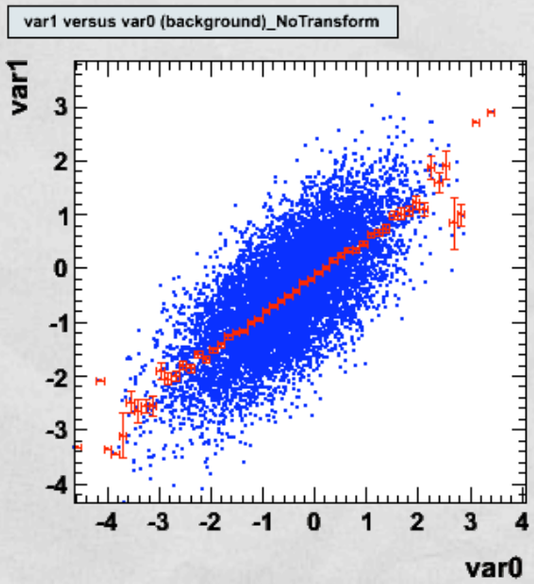


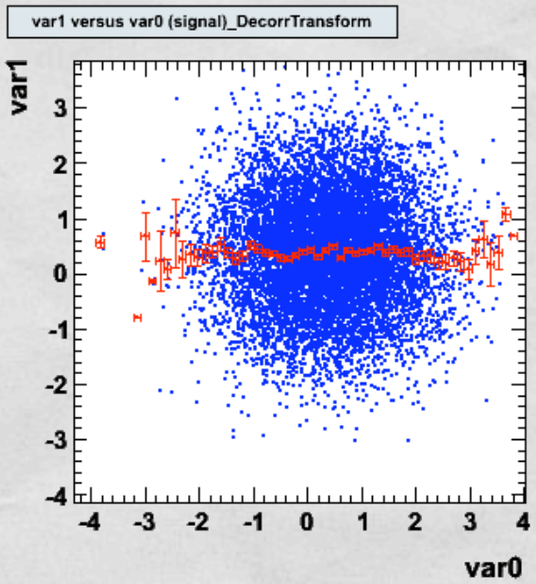


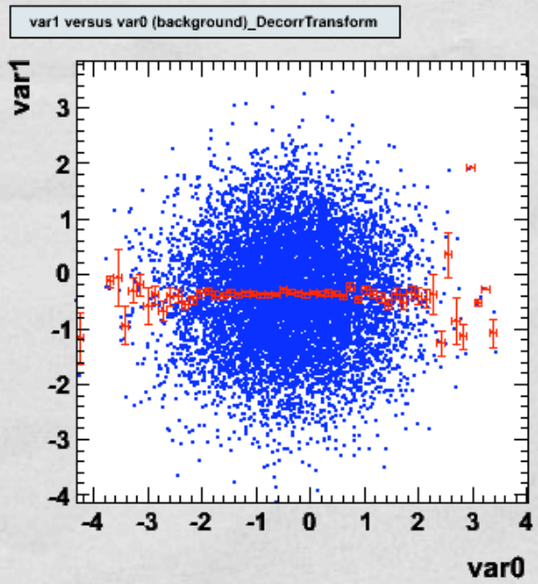


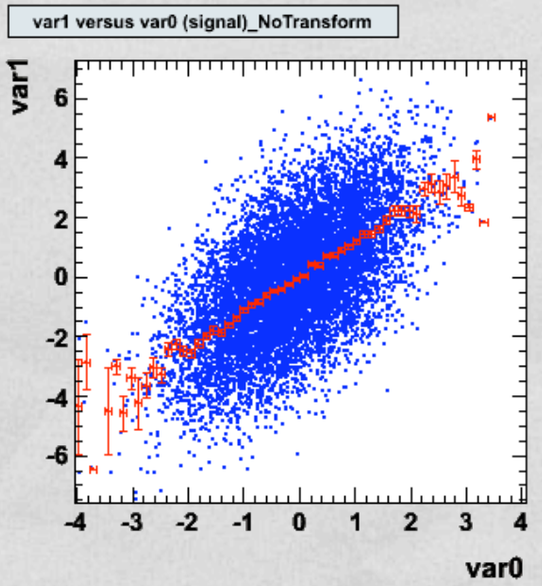
Effetto decorrelazione

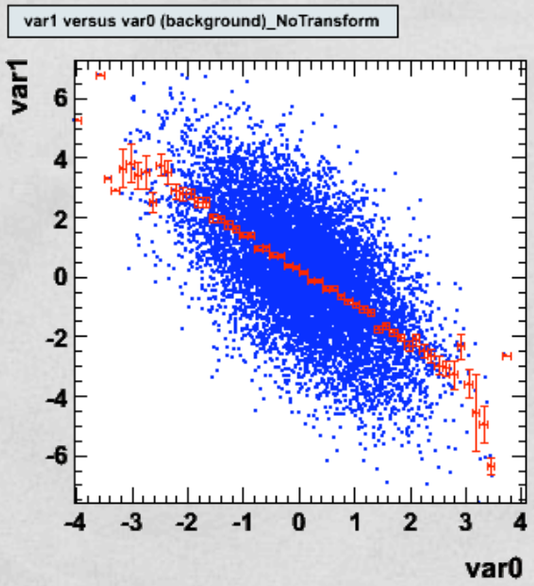


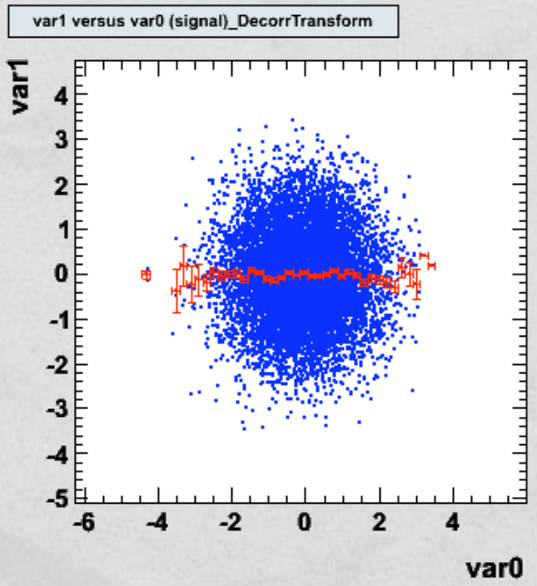


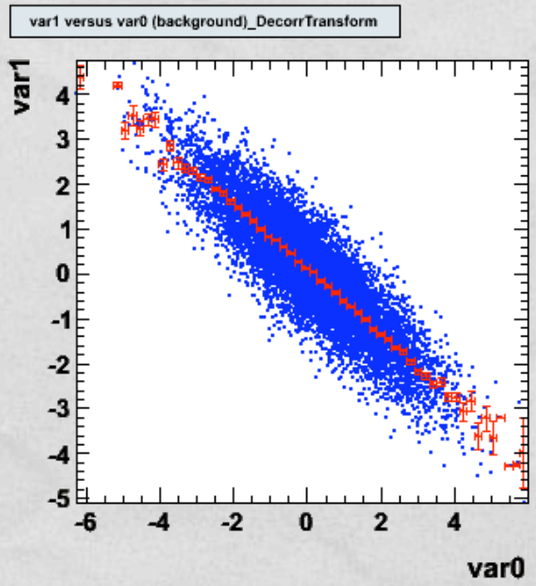


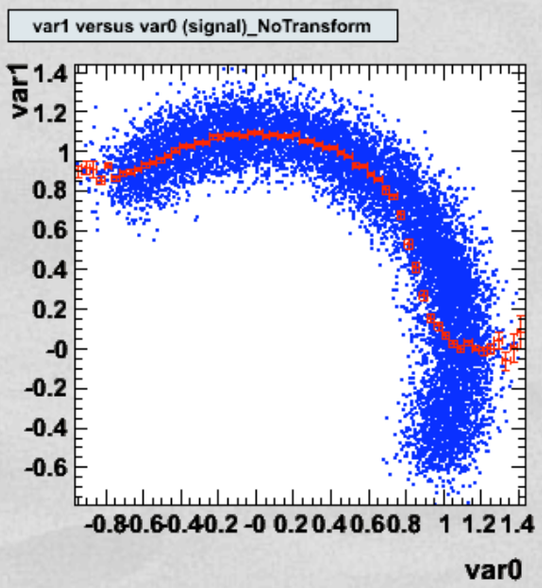


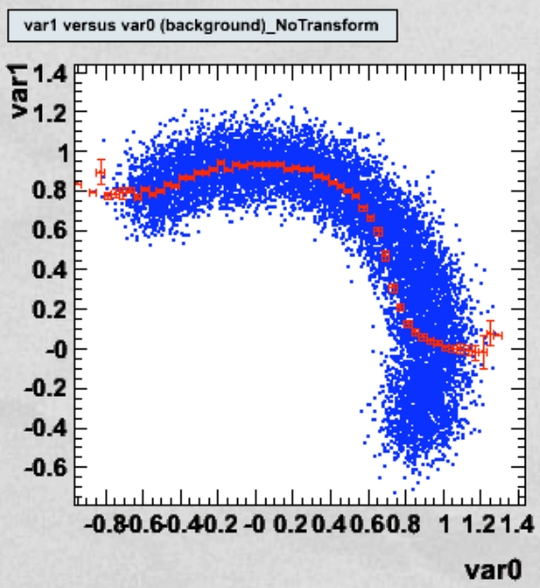


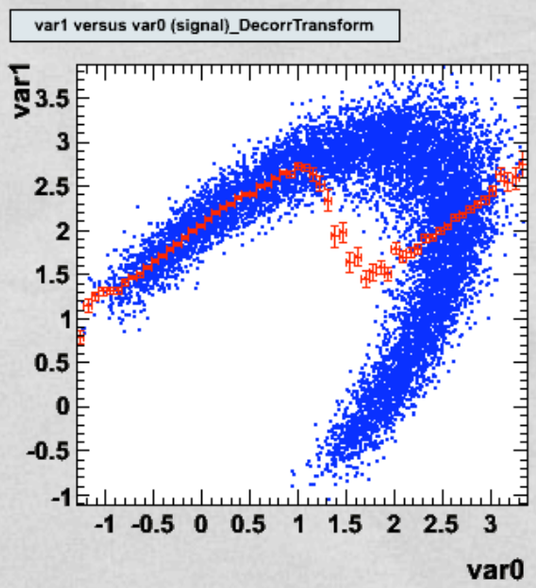


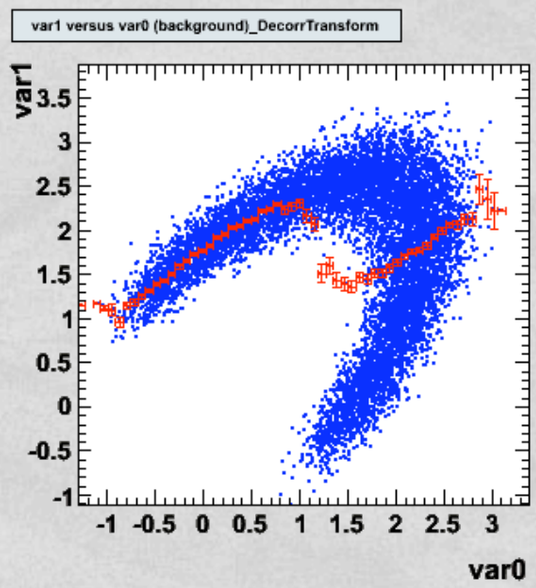








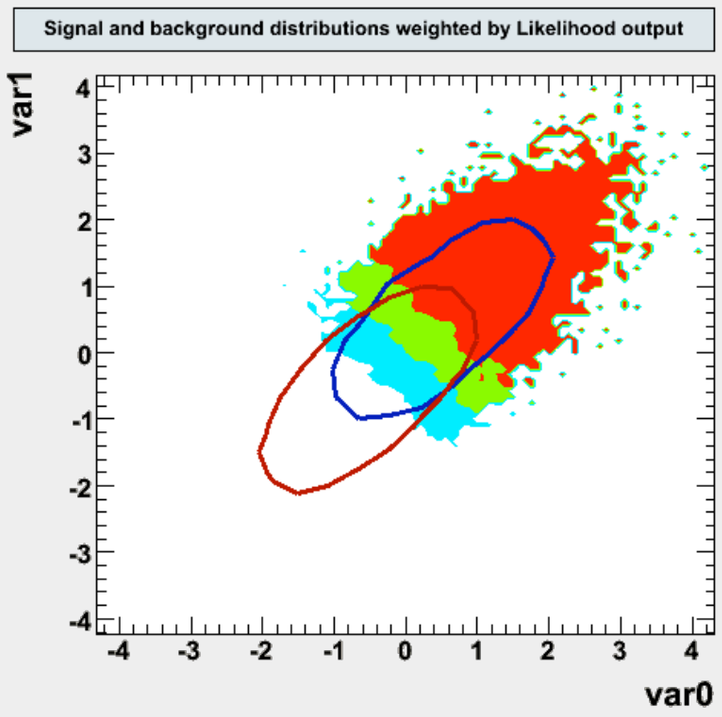




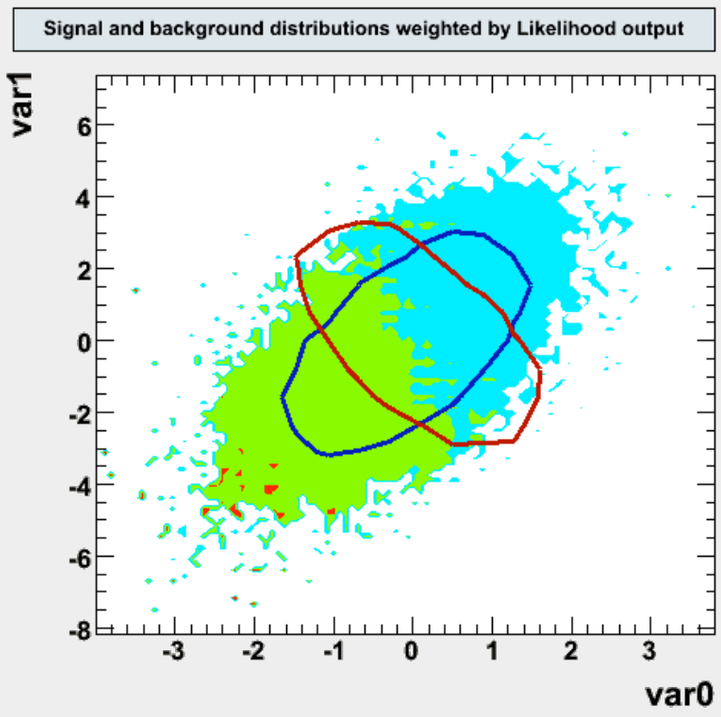
RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

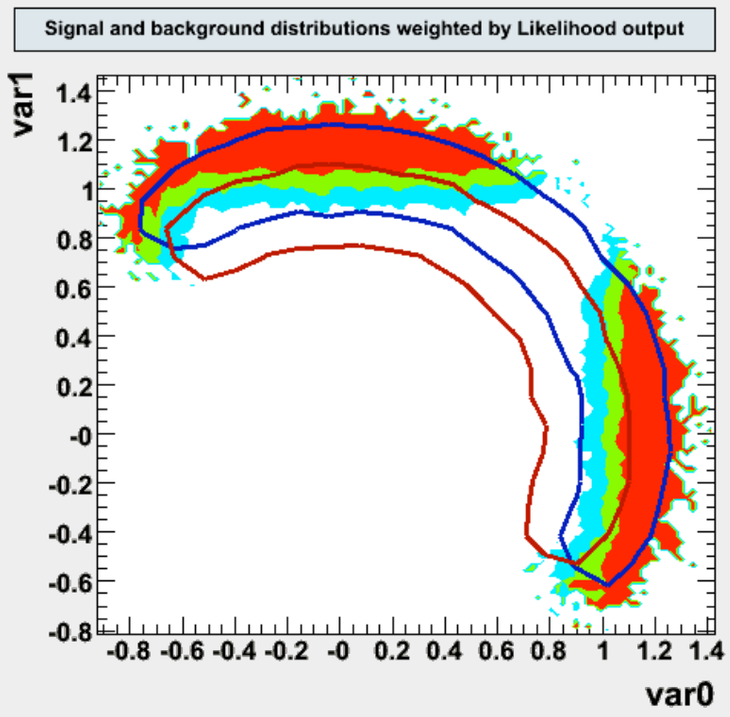
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)



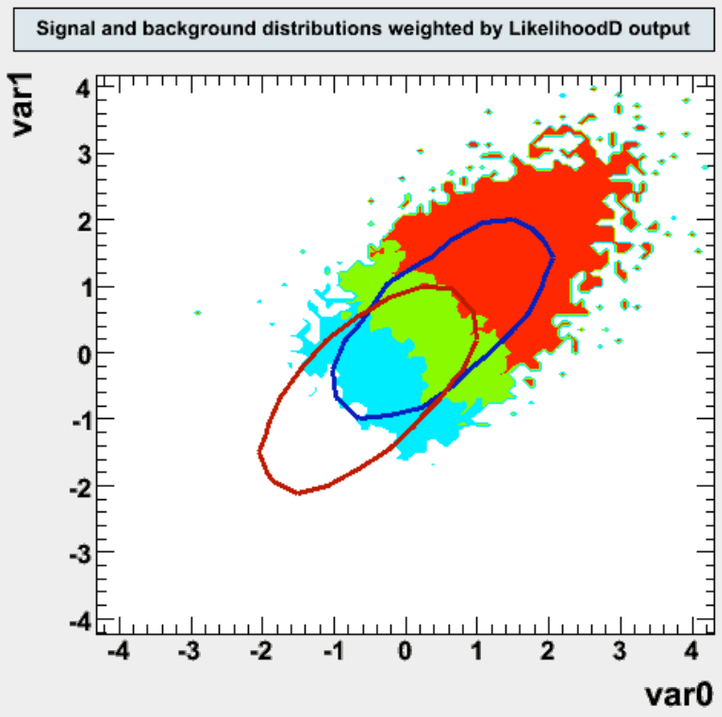
Circular correlations
(same for signal and background)



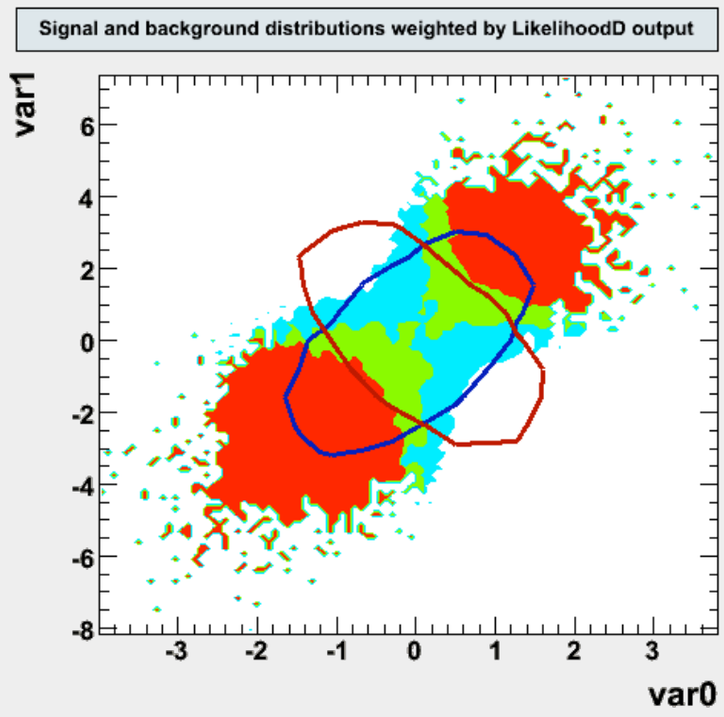
Likelihood

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

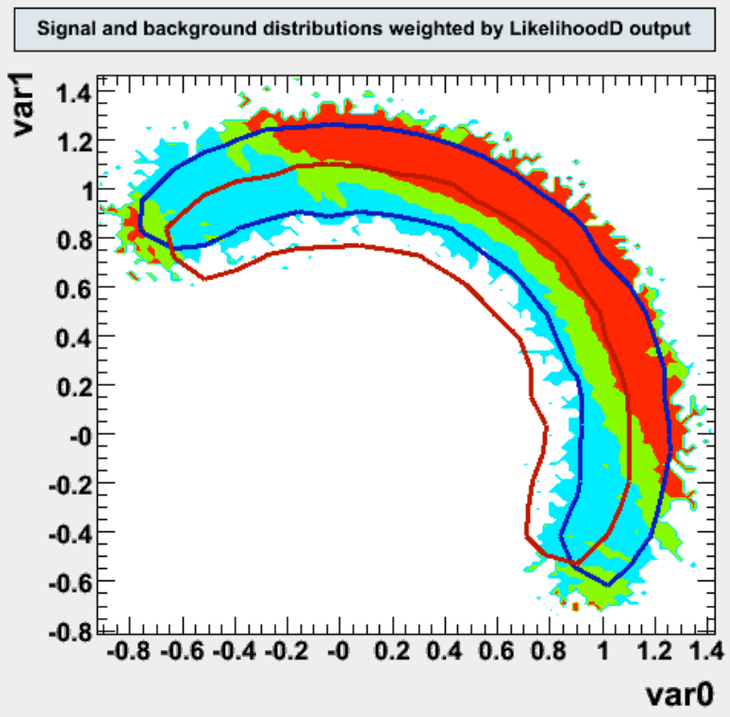
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)



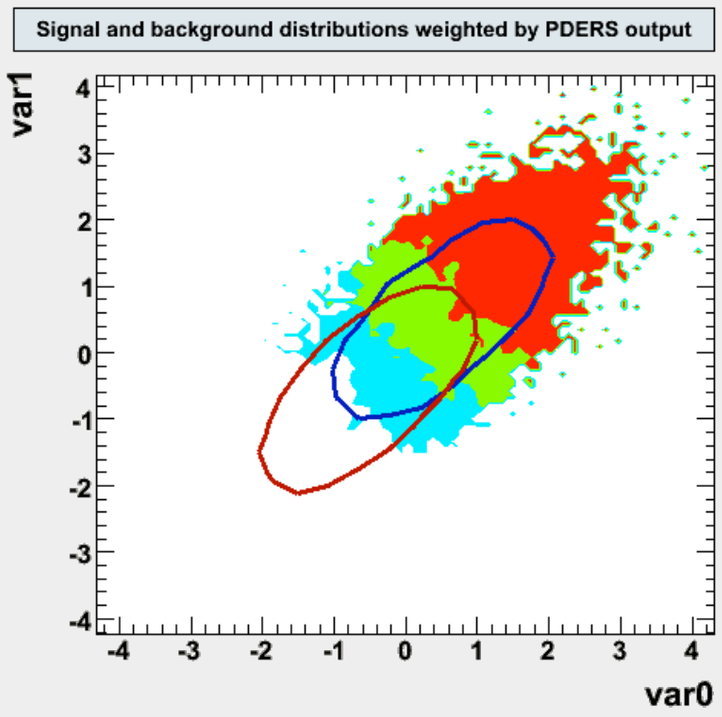
Circular correlations
(same for signal and background)



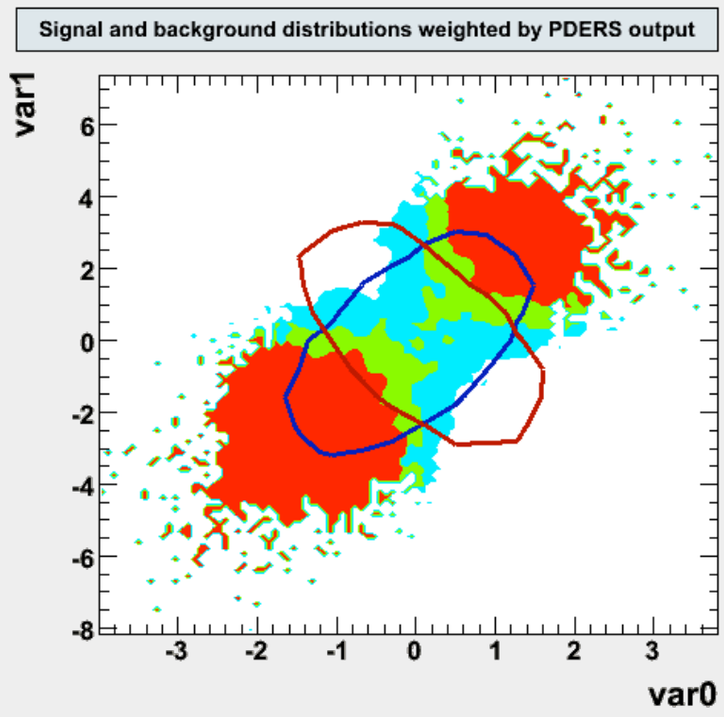
Likelihood - D

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

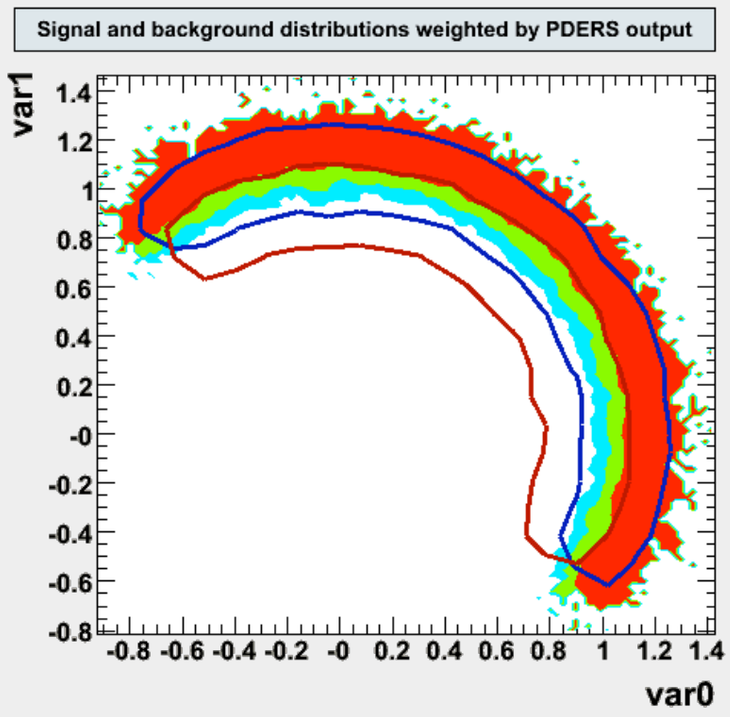
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)



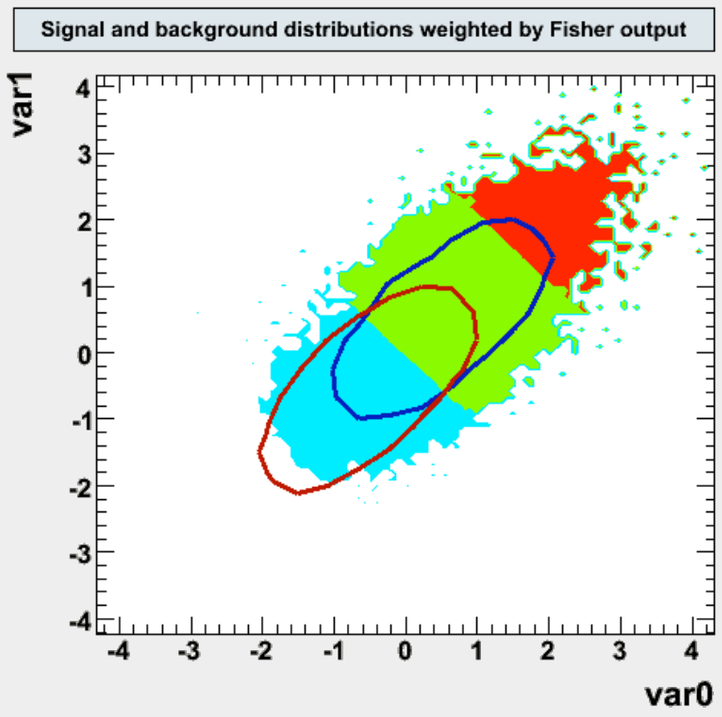
Circular correlations
(same for signal and background)



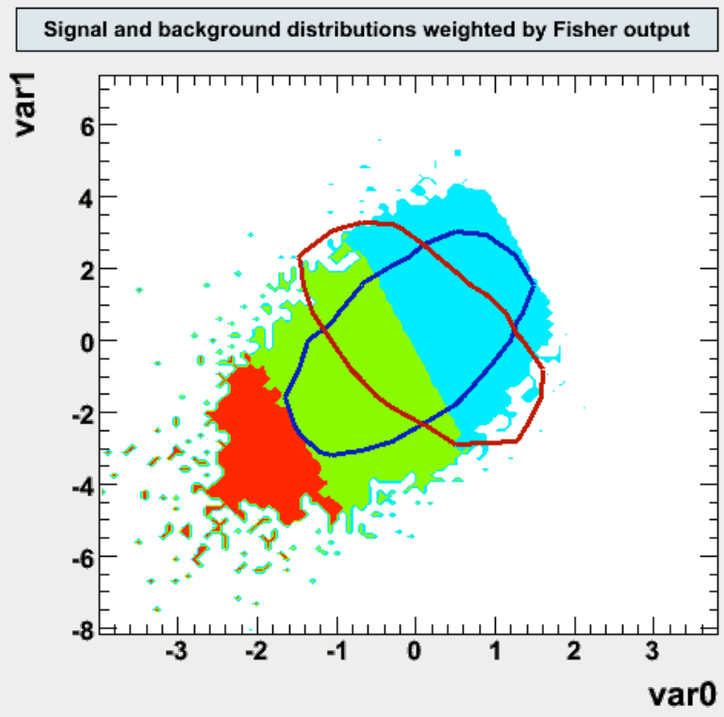
PDERS

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

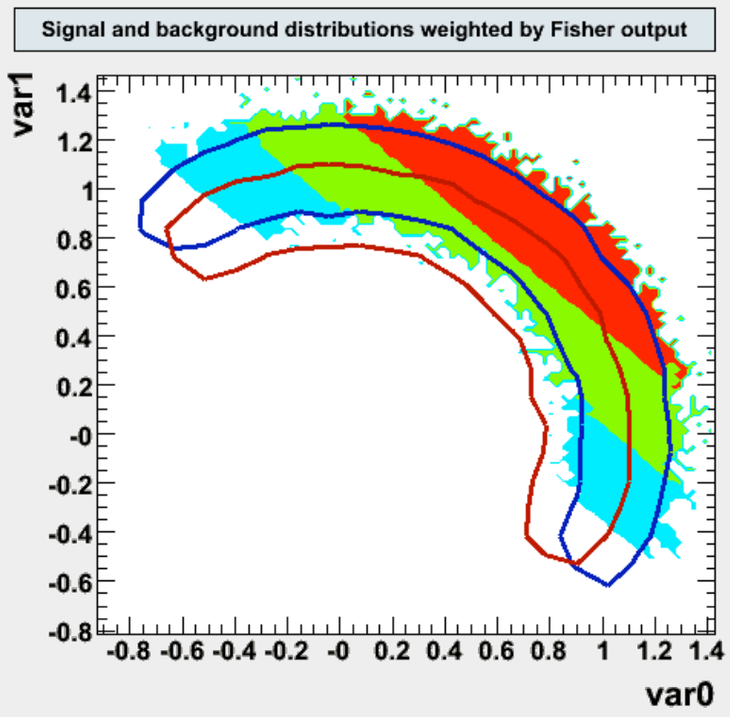
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)



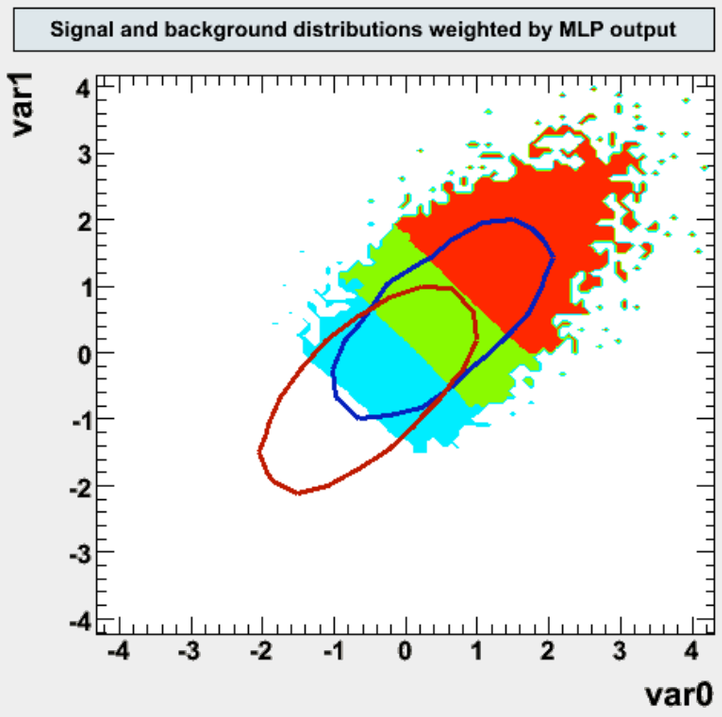
Circular correlations
(same for signal and background)



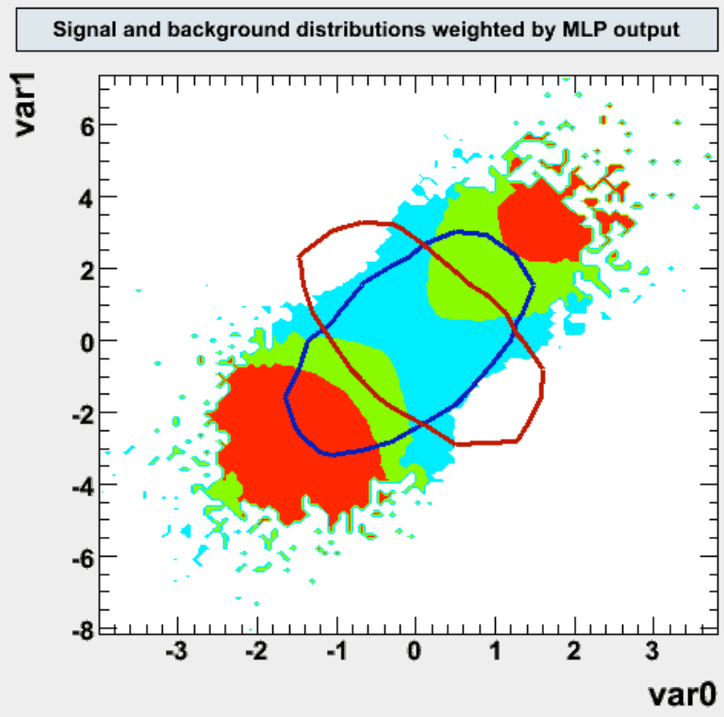
Fisher

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

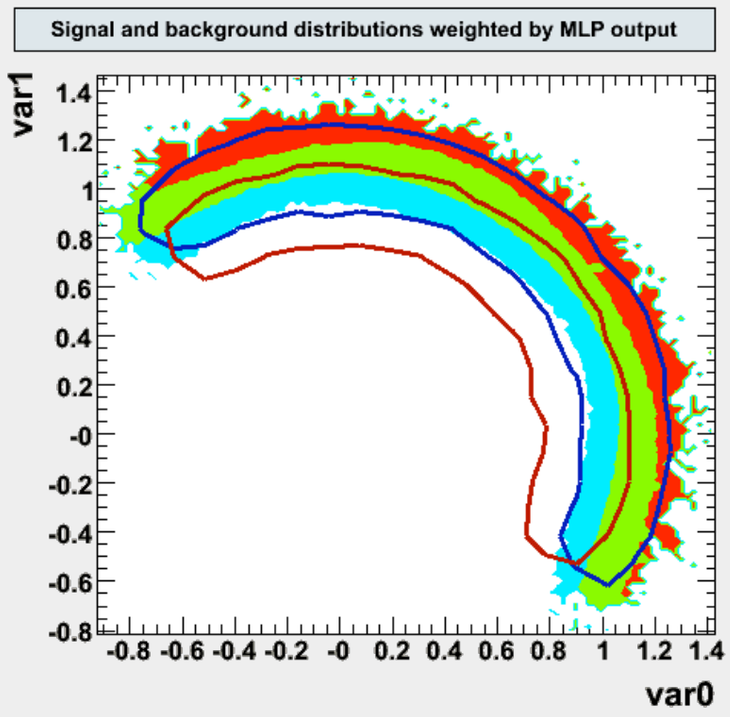
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)



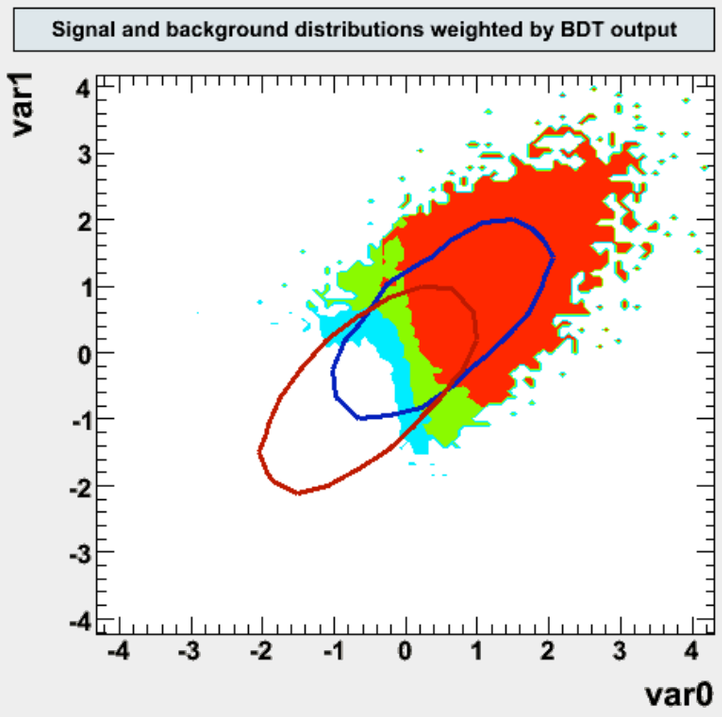
Circular correlations
(same for signal and background)



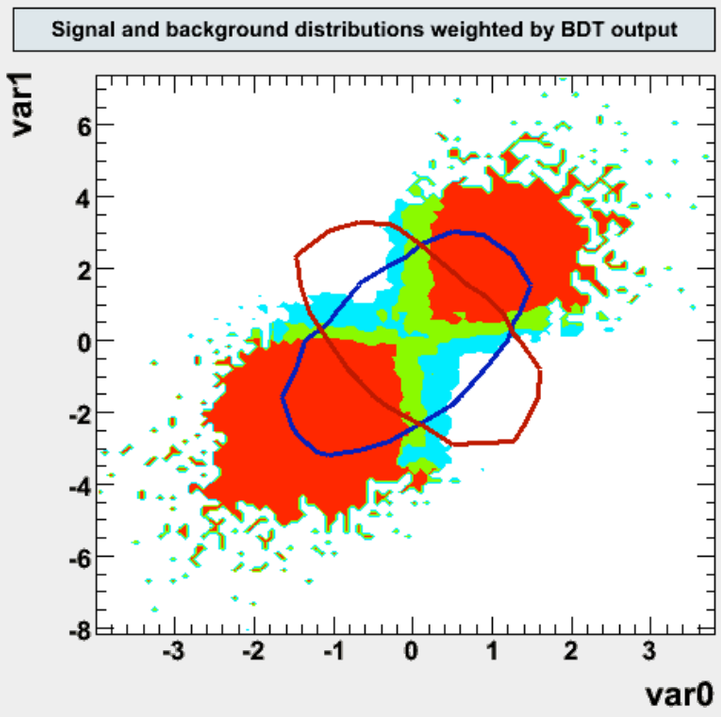
MLP

RICOSTRUZIONE DEI PATTERN DI CORRELAZIONE

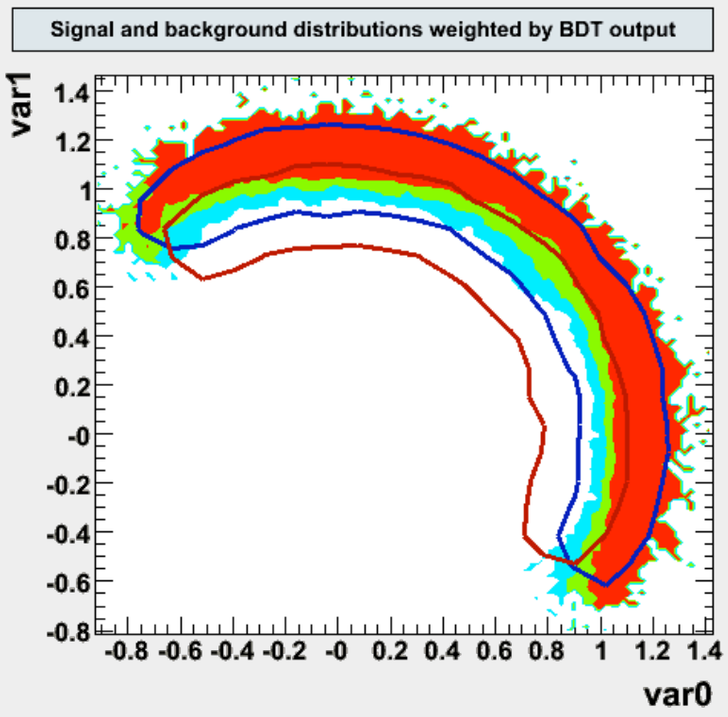
Linear correlations
(same for signal and background)



Cross-linear correlations
(opposite for signal and background)

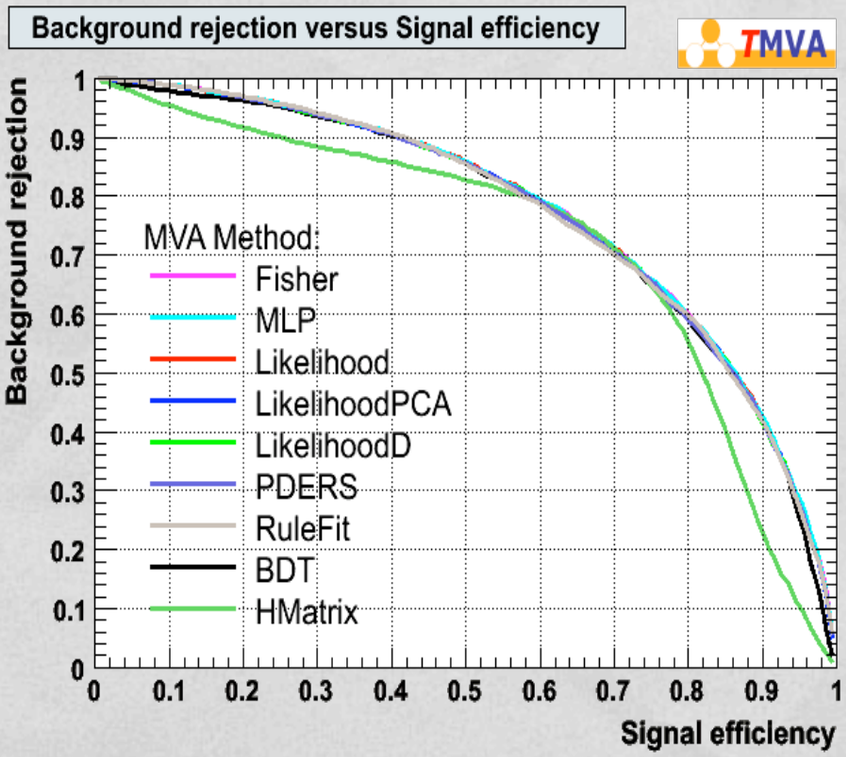


Circular correlations
(same for signal and background)

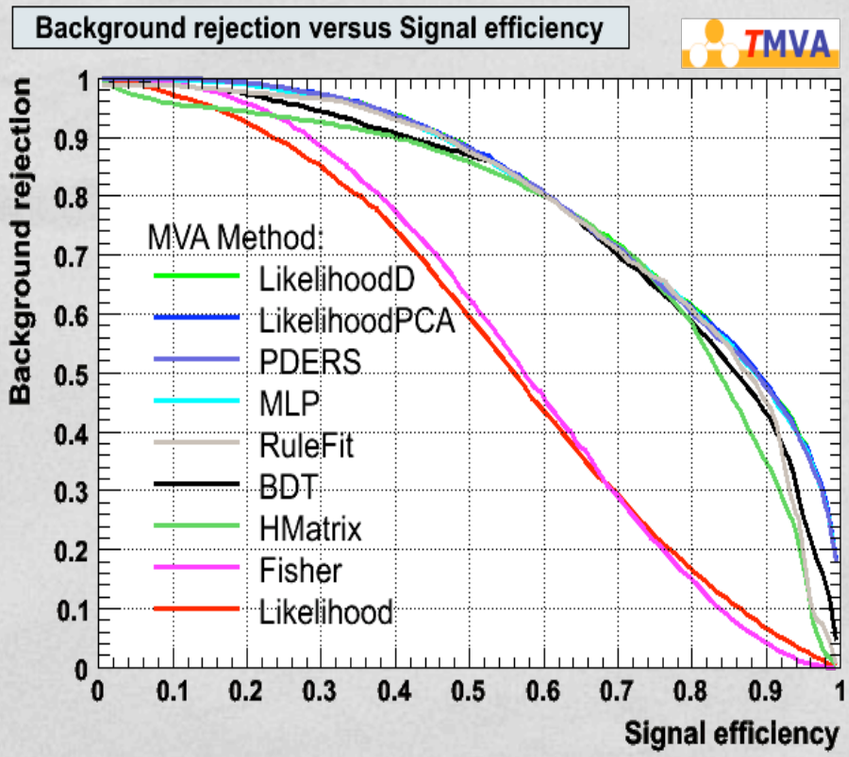


BDT

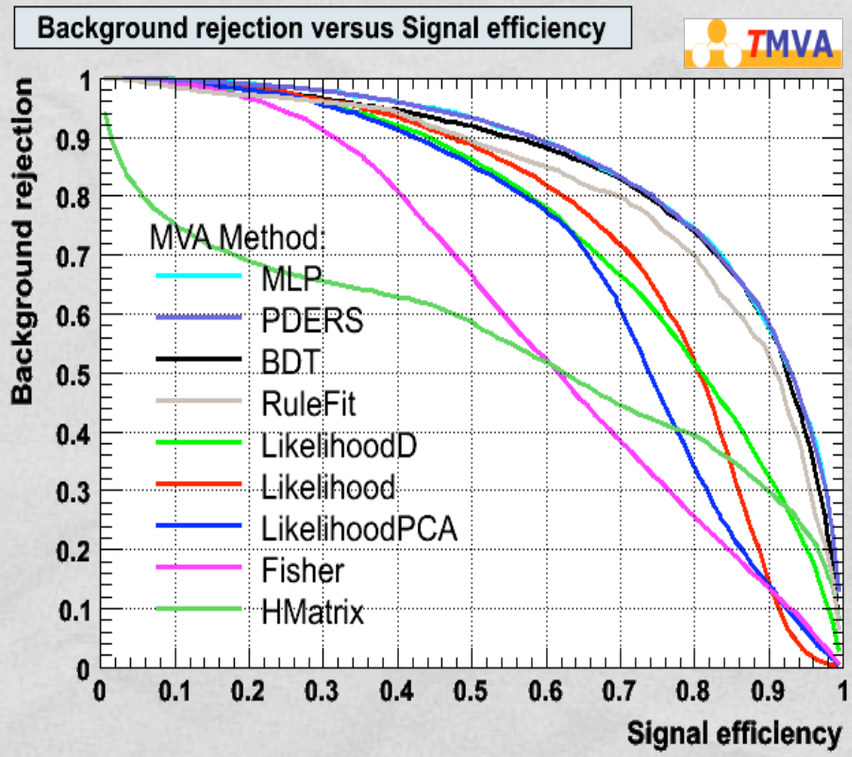
PRESTAZIONI DEI CLASSIFICATORI



Linear Example

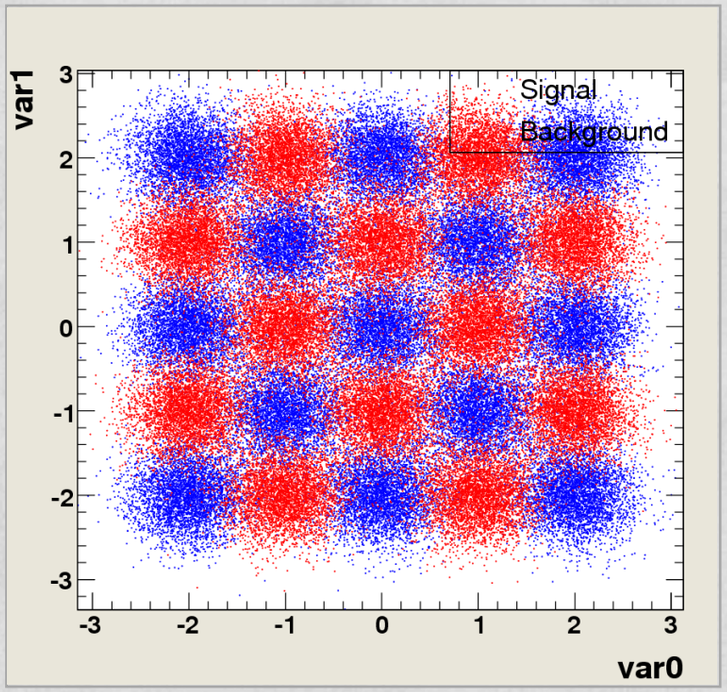


Cross Example

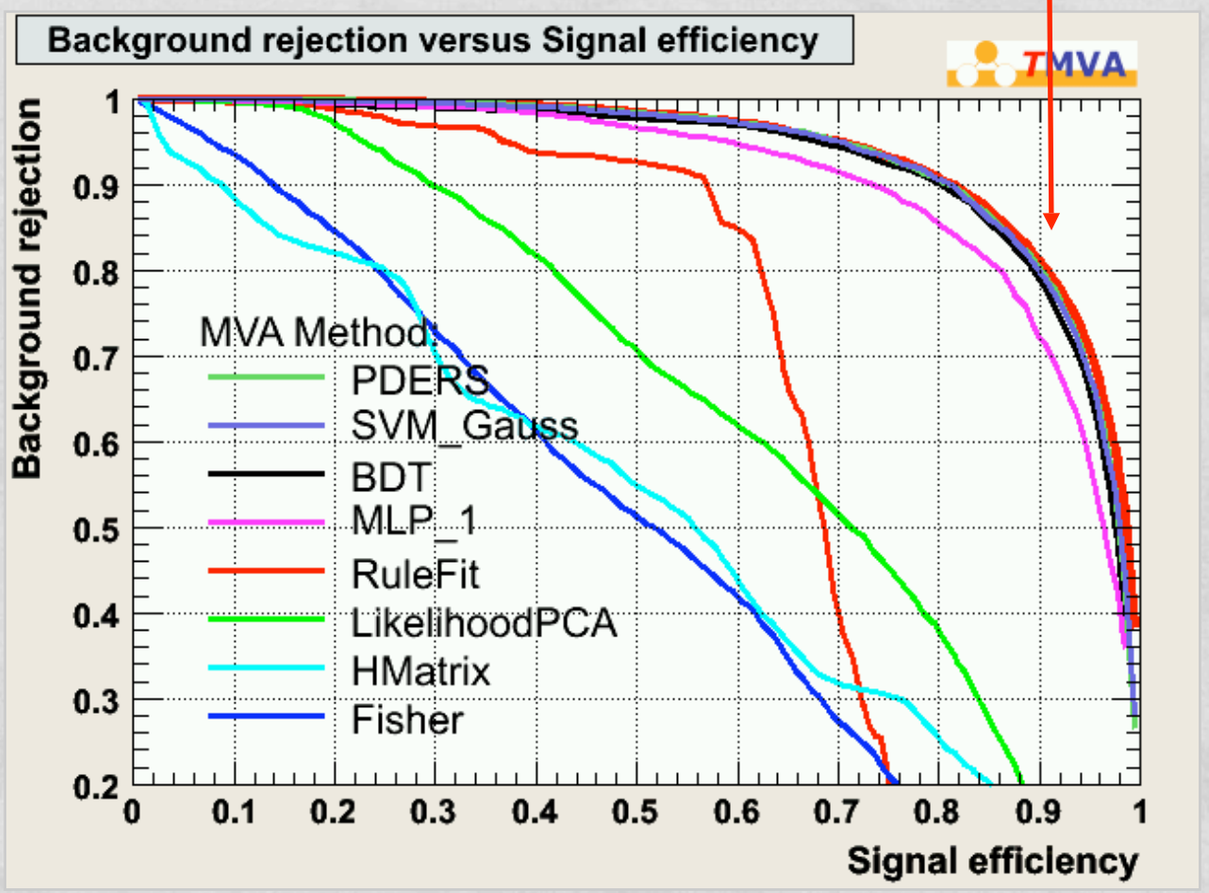


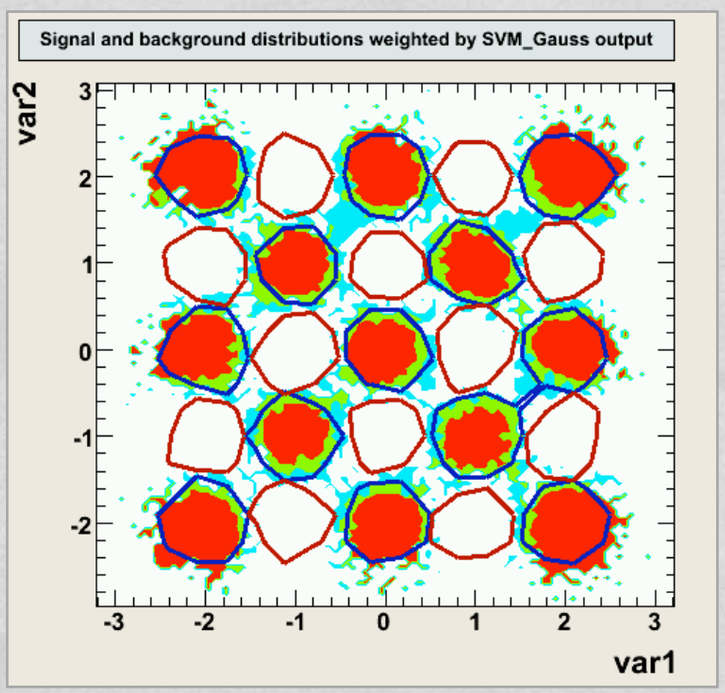
Circular Example

ESEMPIO 2: SCACCHI



Eventi input



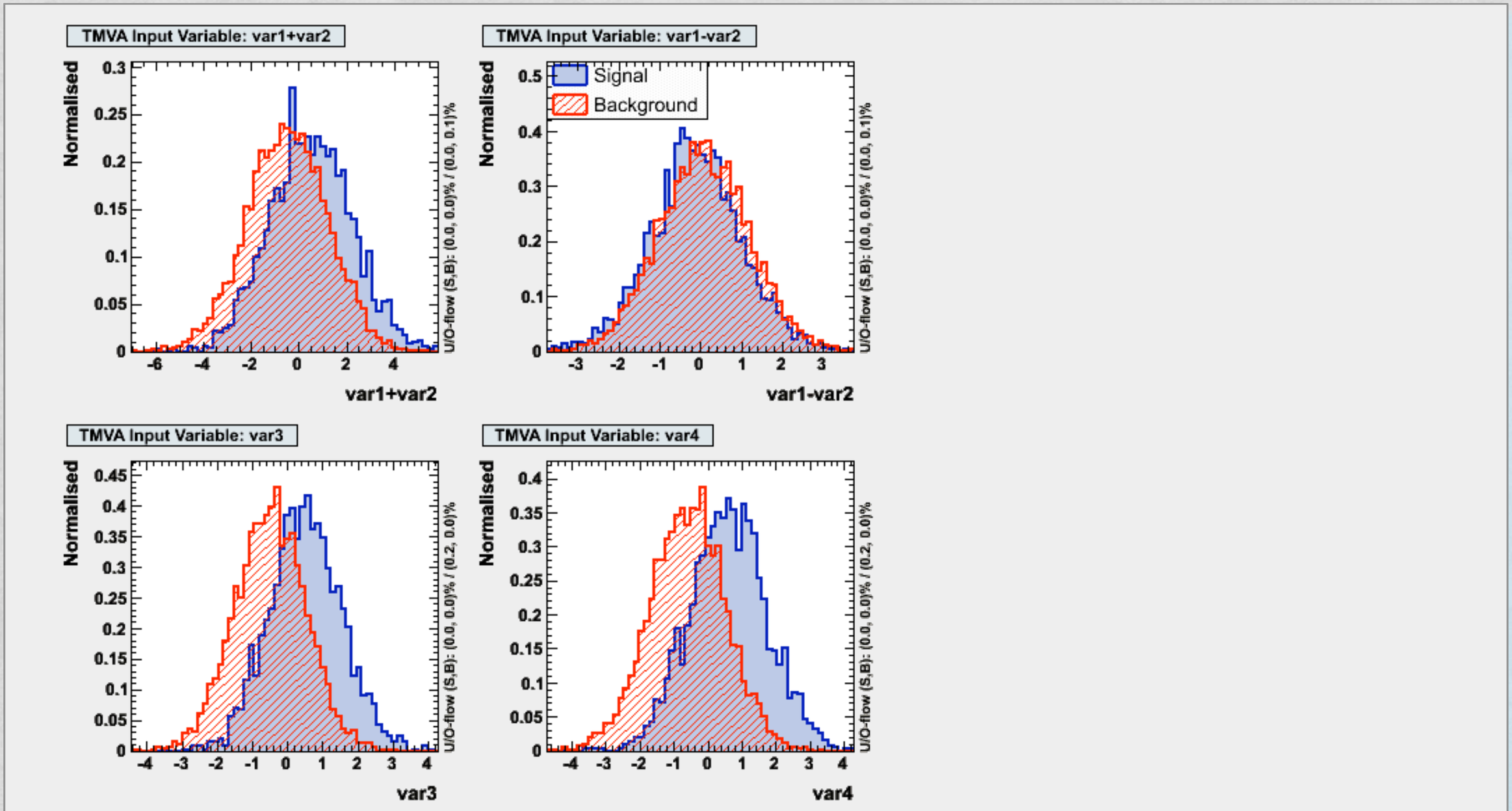


Eventi pesati con risposta SVM

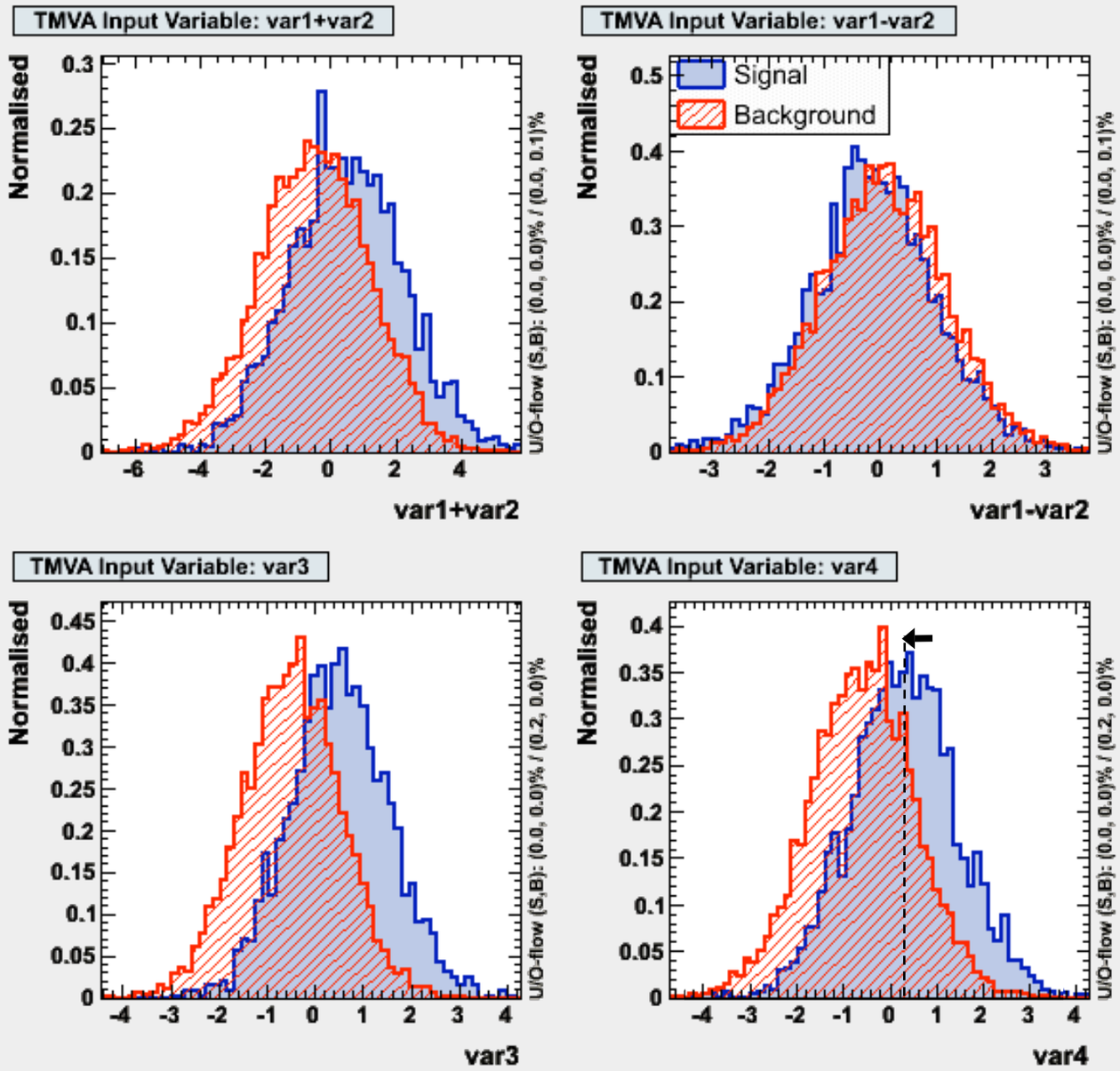
INCERTEZZE SISTEMATICHE

- Cosa succede se una delle variabili di input e' affetta da incertezze di natura sistematica?
- Esempio: una variabile ha maggior potere di separazione nel MC che nei dati rali
  - prestazioni del classificatore sui dati reali < di quelle ottenute in fase di training → bad training
- Soluzioni al problema:
  1. eliminare la variabile: → OK ma perdita di prestazioni
  2. ignorare il problema e valutare le incertezze sistematiche sull'output del classificatore:
     - OK ma classificatore sub-ottimale
  3. training con variabile ripesata/smeared (in accordo ai dati reali o all'effetto sistematico aspettato)
     - OK + classificatore ottimale

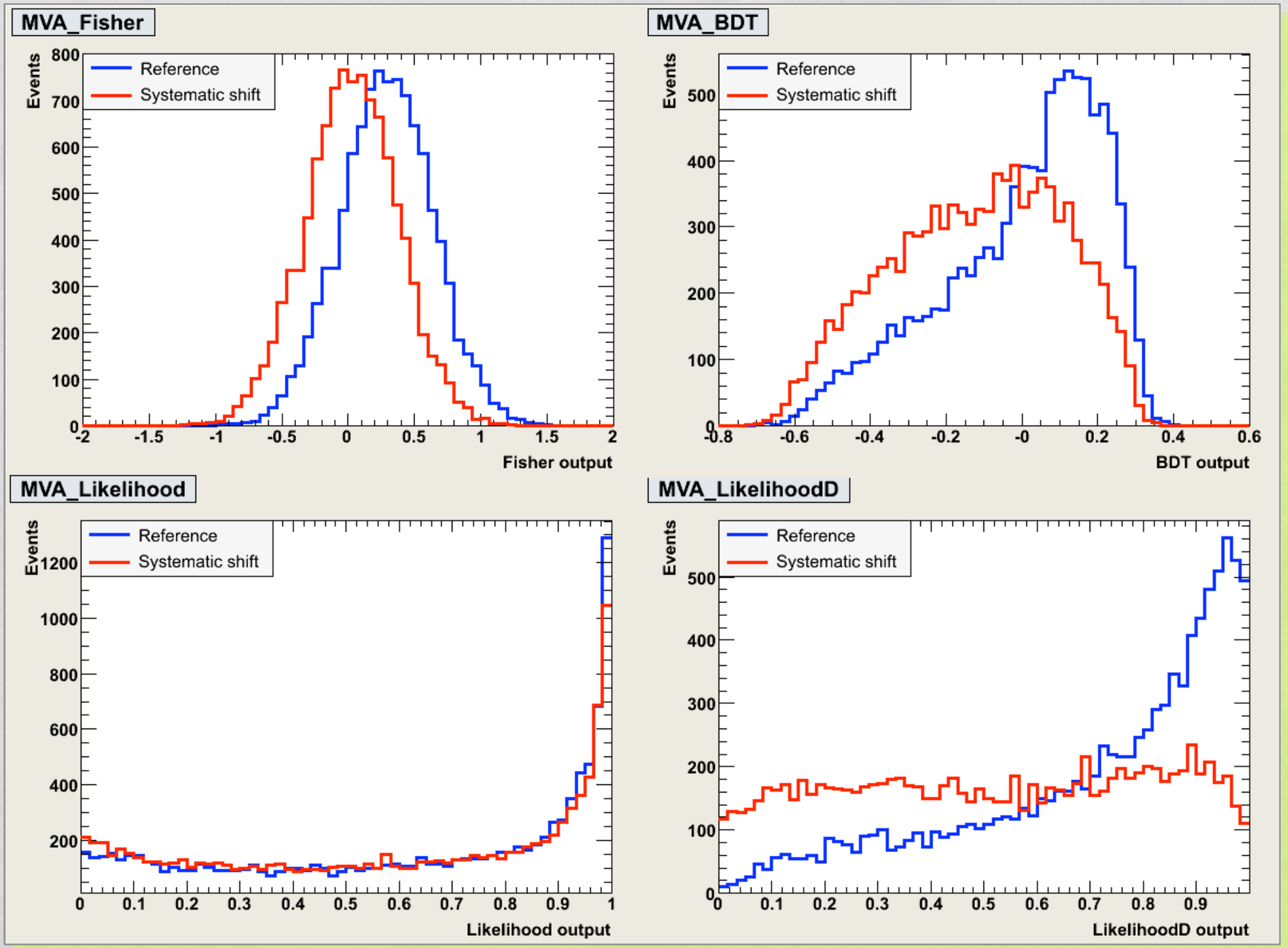
ESEMPIO: var4 affetta da sistematica



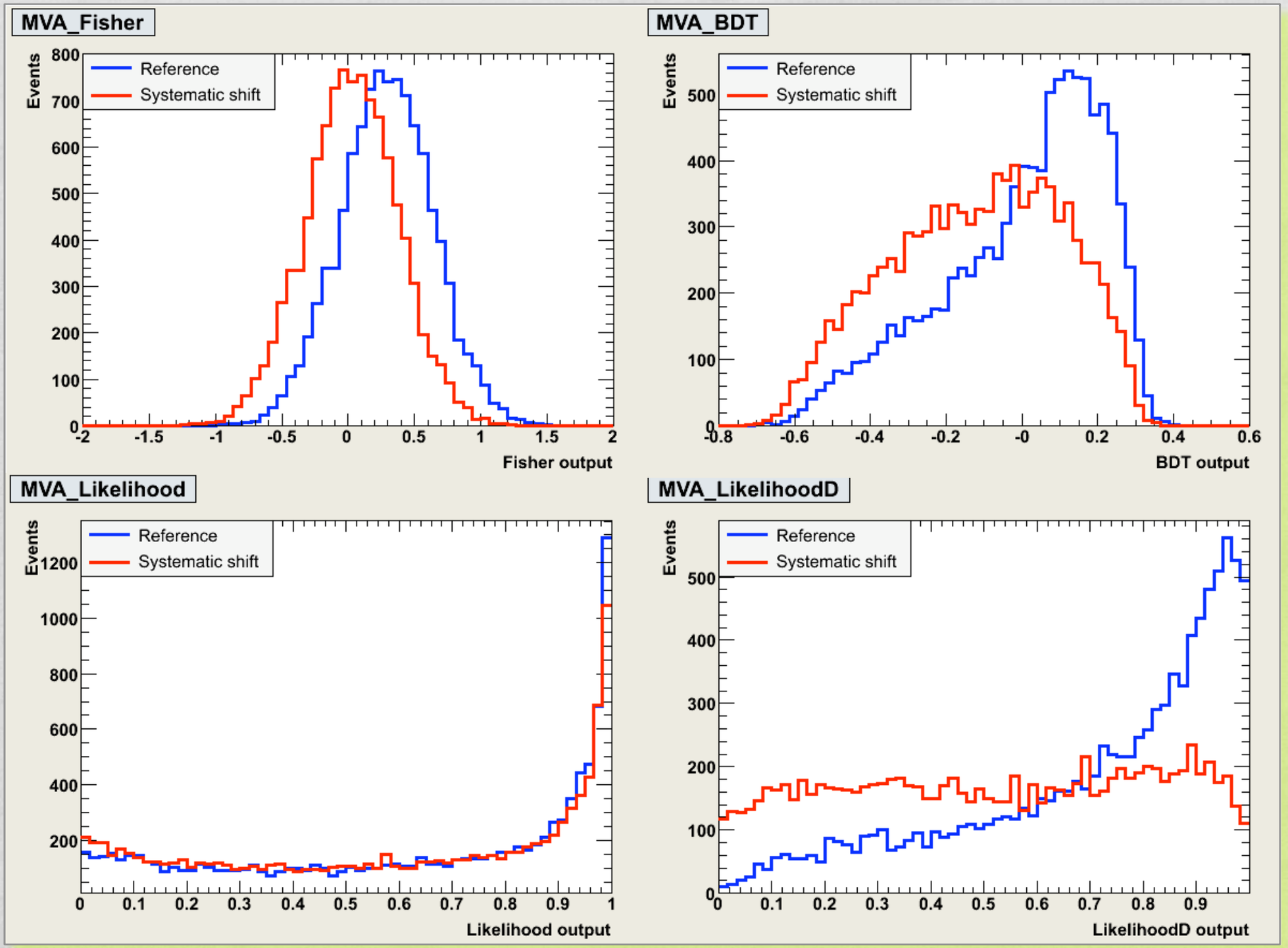
ESEMPIO: var4 affetta da sistematica



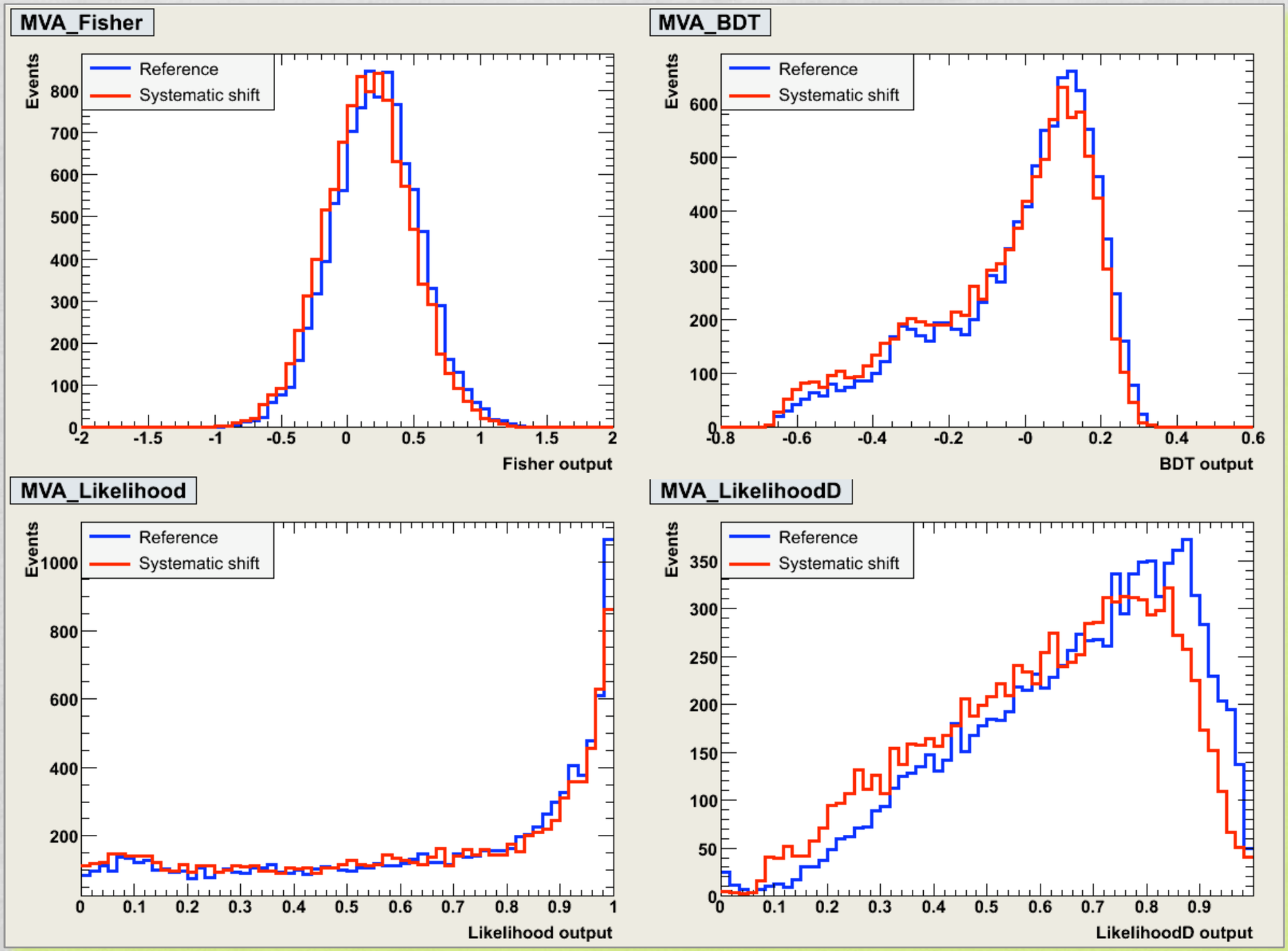
"incertezze di calibrazione" possono spostare il valore centrale e quindi peggiorare il potere di separazione di var4


MVA_Fisher
Reference
Systematic shift
Events
Fisher output
MVA_BDT
Reference
Systematic shift
Events
BDT output
MVA_Likelihood
Reference
Systematic shift
Events
Likelihood output
MVA_LikelihoodD
Reference
Systematic shift
Events
LikelihoodD output

metodo 2


MVA_Fisher
Reference
Systematic shift
Events
Fisher output
MVA_BDT
Reference
Systematic shift
Events
BDT output
MVA_Likelihood
Reference
Systematic shift
Events
Likelihood output
MVA_LikelihoodD
Reference
Systematic shift
Events
LikelihoodD output

metodo 3


MVA_Fisher
Reference
Systematic shift
Events
Fisher output
MVA_BDT
Reference
Systematic shift
Events
BDT output
MVA_Likelihood
Reference
Systematic shift
Events
Likelihood output
MVA_LikelihoodD
Reference
Systematic shift
Events
LikelihoodD output