LMS:

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \eta^n e_n \underline{x}^n \,;\eta^n = \frac{\lambda}{\left\|\underline{x}^n\right\|^2}$$ 0<λ<2. λ=1 implies projection.

Convergence of LMS: Convergence in the mean; convergence in mean square

Momentum: $\underline{w}^{(n+1)} = \underline{w}^{(n)} + \eta(z^n - \underline{w}^{(n)T}\underline{x}^n)\underline{x}^n + \mu(\underline{w}^{(n)} - \underline{w}^{(n-1)})$

$$\left.\begin{array}{l} \Delta\underline{w}^{(n)} = \mu\Delta\underline{w}^{(n-1)} + \eta e_n \underline{x}^n \\ or\ \underline{d}^{(n)} = \mu\underline{d}^{(n-1)} - \eta\underline{g}^{(n)} \\ \text{clearly need } \mu < 1 \end{array}\right\}$$

Nesterov: $\underline{d}^{(n)} = \mu\underline{d}^{(n-1)} - \eta\underline{g}^{(n)}\,|_{\underline{w}^{(n)}+\mu\underline{d}^{(n-1)}}$ ...*gradient evaluated at* $\underline{w}^{(n)} + \mu\underline{d}^{(n-1)}$

Bold-driver: $\eta^{new} = \begin{cases} \rho\ \eta^{old} & \text{if } \Delta J < 0 \ \Rightarrow \text{ improved} \\ \sigma\ \eta^{old} & \text{if } \Delta J > 0 \end{cases} \quad \rho = 1.1 \quad \sigma \approx 0.5$

AdaGrad:

$$w_i^{(n+1)} = w_i^{(n)} - \eta_i^{(n)} g_i^{(n)};\eta_i^{(n)} = \frac{\eta}{\sqrt{\sum_{j=1}^{n}\left(g_i^{(j)}\right)^2 + \varepsilon}} = \frac{\eta}{\sqrt{G_n + \varepsilon}};\varepsilon \approx 10^{-8}$$

$$G_n = G_{n-1} + \left(g_i^{(n)}\right)^2;G_0 = 0$$

RMSprop:

$$G_n = \gamma G_{n-1} + (1-\gamma)\,\|\,\underline{g}^{(n)}\,\|_2^2;G_0 = 0;\gamma \approx 0.9$$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \frac{\eta}{\sqrt{G_n + \varepsilon}}\underline{g}^n;\eta \approx 0.001$$

Adam:

$$\overline{g}^{(n)} = \theta\,\overline{g}^{(n-1)} + (1-\theta)g^{(n)};\overline{g}^{(0)} = 0$$

$$G_n = \gamma G_{n-1} + (1-\gamma)\,\|\,\underline{g}^{(n)}\,\|_2^2$$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \frac{\eta^{(n)}}{\sqrt{G_n} + \varepsilon\sqrt{1-\gamma^t}}\underline{\overline{g}}^n;\eta^{(n)} = \eta\frac{\sqrt{1-\gamma^t}}{1-\theta^t}$$

Quick prop: $w_i^{(n+1)} - w_i^{(n)} = \frac{g_i^{(n)}}{g_i^{(n-1)} - g_i^{(n)}}\left[w_i^{(n)} - w_i^{(n-1)} \right]$

$$Idea: Parabola: aw_i^2 + bw_i + c \Rightarrow \min at\ w_i = -\frac{b}{2a}$$
$$2aw_i^{(n-1)} + b = g_i^{(n-1)}$$
$$2aw_i^{(n)} + b = g_i^{(n)} \ ......\ (1)$$
$$\Rightarrow 2a = \frac{g_i^{(n)} - g_i^{(n-1)}}{w_i^{(n)} - w_i^{(n-1)}} \Rightarrow w_i^{(n+1)} = -\frac{b}{2a} = w_i^{(n)} - \frac{g_i^{(n)}}{2a} \ldots \text{from (1)}$$
$$\Rightarrow w_i^{(n+1)} = w_i^{(n)} + \frac{g_i^{(n)}}{g_i^{(n-1)} - g_i^{(n)}}\left(w_i^{(n)} - w_i^{(n-1)} \right) \Rightarrow w_i^{(n+1)} = \frac{g_i^{(n-1)} w_i^{(n)} - g_i^{(n)} w_i^{(n-1)}}{g_i^{(n-1)} - g_i^{(n)}}$$

Single layer network: Nonlinearity has a local effect…. This is exploited in MLP

$$\hat{z}^n = g[y(\underline{w}, \underline{x}^n)] = g(\underline{w}^T \underline{x}^n) = \frac{1}{1 + e^{-\underline{w}^T \underline{x}^n}}$$
$$\nabla J(\underline{w}) = -\sum_{n=1}^{N} (z^n - \hat{z}^n) \nabla \hat{z}^n$$
$$= -\sum_{n=1}^{N} e_n \cdot \underbrace{g'}_{\substack{\text{local gradient}\\ \text{of neuron}}} \underline{x}^n$$

Incremental Newton and RLS

Recall Information matrix: $\Sigma^{(n)-1} = \Sigma^{(n-1)-1} + \underline{x}^n \left(\underline{x}^n \right)^T$

From MIL: $\Sigma^{(n)} = \Sigma^{(n-1)} - \frac{\Sigma^{(n-1)} \underline{x}^n \left(\underline{x}^n \right)^T \Sigma^{(n-1)}}{1 + \left(\underline{x}^n \right)^T \Sigma^{(n-1)} \underline{x}^n}$

$$\underline{w}^{(n)} = [\Sigma^{(n-1)} - \frac{\Sigma^{(n-1)} \underline{x}^n \left(\underline{x}^n \right)^T \Sigma^{(n-1)}}{1 + \left(\underline{x}^n \right)^T \Sigma^{(n-1)} \underline{x}^n}][\sum_{i=1}^{n-1} \underline{x}^i z^i + \underline{x}^n z^n]$$
$$= \underline{w}^{(n-1)} + \underline{k}^n [z^n - \left(\underline{x}^n \right)^T \underline{w}^{(n-1)}]; \underline{k}^n = \frac{\Sigma^{(n-1)} \underline{x}^n}{1 + \left(\underline{x}^n \right)^T \Sigma^{(n-1)} \underline{x}^n}$$

Discuss algorithm and fading memory

Modified RLS = Gauss-Newton = EKF

$$\underline{w}^{(n-1)}$$

$$z^n = g(\underline{w}^T \underline{x}^n) \approx g((\underline{w}^{(n-1)})^T \underline{x}^n) + g'((\underline{w}^{(n-1)})^T \underline{x}^n)(\underline{x}^n)^T (\underline{w} - \underline{w}^{(n-1)})$$

$$r_n = z^n - g((\underline{w}^{(n-1)})^T \underline{x}^n) + (\underline{\tilde{x}}^n)^T \underline{w}^{(n-1)} = (\underline{\tilde{x}}^n)^T \underline{w}$$

$$Compute\, r_n \text{ and } \underline{\tilde{x}}^n \text{ at sample } n$$

$$\underline{k} \leftarrow \frac{\Sigma^{(n-1)} \underline{\tilde{x}}^n}{1 + \left(\underline{\tilde{x}}^n\right)^T \Sigma^{(n-1)} \underline{\tilde{x}}^n}$$

$$\underline{w}^{(n)} = \underline{w}^{(n-1)} + \underline{k}(r_n - (\underline{\tilde{x}}^n)^T \underline{\hat{w}}^{(n-1)}) \Rightarrow \underline{w}^{(n)} = \underline{w}^{(n-1)} + \underline{k}(z^n - g((\underline{w}^{(n-1)})^T \underline{x}^n)$$

For logistic: $\underline{\tilde{x}}^n = g(\underline{w}^{(n-1)T} \underline{x}^n)[1 - g(\underline{w}^{(n-1)T} \underline{x}^n)]\underline{x}^n = \hat{z}^n(1-\hat{z}^n)\underline{x}^n$

Fisher's Linear Discriminant:

$$S_T = S_W + S_B$$

$$S_T = \sum_{n=1}^{N} (\underline{x}^n - \underline{\mu})(\underline{x}^n - \underline{\mu})^T ; \underline{\mu} = \frac{1}{N}\sum_{n=1}^{N} \underline{x}^n$$

$$S_W = \sum_{k=1}^{C} S_k ; S_k = \sum_{\substack{n=1 \\ n:z^n=k}}^{N} (\underline{x}^n - \underline{\mu}_k)(\underline{x}^n - \underline{\mu}_k)^T ; \underline{\mu}_k = \frac{\sum_{\substack{n=1 \\ n:z^n=k}}^{N} \underline{x}^n}{N_k}$$

$$N_k = \sum_{n=1}^{N} \delta_{z^n k} ; \delta_{z^n k} = Kronecker\ delta\ function$$

$$S_B = \sum_{k=1}^{C} N_k (\underline{\mu}_k - \underline{\mu})(\underline{\mu}_k - \underline{\mu})^T$$

$$Find\ discriminant\ functions\ \underline{g} = W^T \underline{x} \ni tr\{\left(WS_W W^T\right)^{-1}\left(WS_B W^T\right)\} \text{ is maximum}$$

$$\Rightarrow \text{Normalized Eigen vectors of } (S_W^{-1} S_B) \text{ corresponding to } (C\text{-}1) \text{ largest eigen values}$$

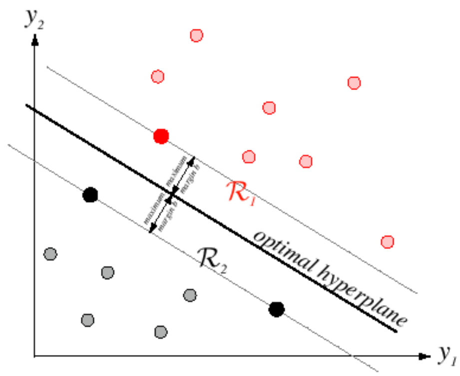
PCA versus LDA

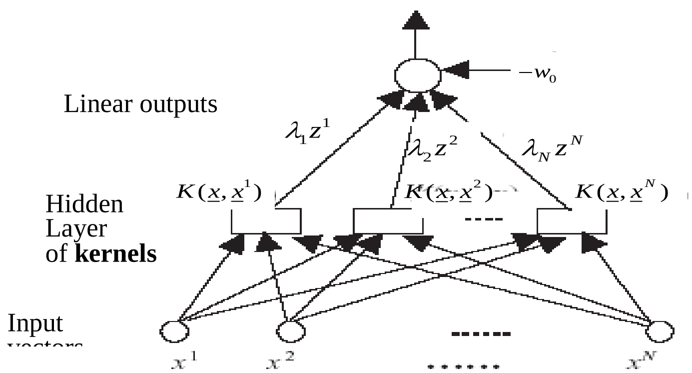
PCA: Dimensionality reduction while preserving as much of the variance in the high dimensional space as possible.
LDA: Dimensionality reduction while preserving as much of the class discriminatory information as possible.

Key Idea of SVM: Nonlinearly transforms data into a higher dimensional feature space such that the classes are linearly separable and finds an optimal hyperplane separating each pair of classes in the new space

Can we find a hyperplane with the largest separation (margin) between two classes? … Large margin classifier

SVM formulates the problem of finding the largest margin as a quadratic programming problem. It maximizes the distance from the nearest training patterns. Excellent Method.





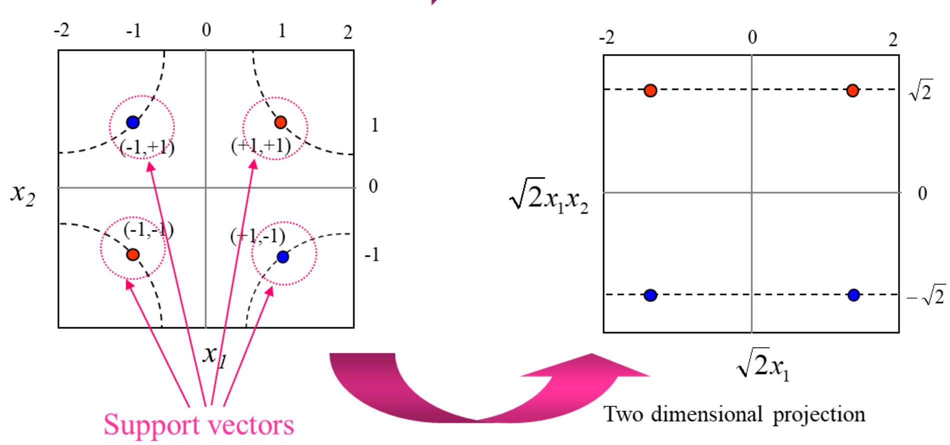
Kernels allow you to transform data for linear separability. Kernels exploit inner product between data points.

$K = [K(\underline{x}^i, \underline{x}^j)]$ *is a Kernel if* $K \geq 0$

Mercer's theorem

SVM and XOR: How transformation of data can make a linear classifier work!

$$x_1, x_2 \rightarrow 1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2$$



Discuss Minimum distance from a point $\underline{x}$ to a hyperplane: $\underline{w}^T \underline{x}_p - w_0 = 0$ Discuss duality.

$$\| \underline{x} - \underline{x}_p \|_2 = \left(\frac{| \underline{w}^T \underline{x} - w_0 |}{\underline{w}^T \underline{w}} \right) \| \underline{w} \|_2$$

$$\underline{x} = \underline{0} \Rightarrow \| \underline{x}_p \|_2 = \frac{| w_0 |}{\| \underline{w} \|_2} \quad \boxed{\textit{Distance of the plane from the origin}}$$

QP problem: $$\min_{\underline{w}} \quad \frac{1}{2} \underline{w}^T \underline{w}$$
$$z^i (\underline{w}^T \underline{x}^i - w_0) - 1 \geq 0 \quad \forall i$$

Dual: $$q(\underline{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j z^i z^j \left(\underline{x}^i \right)^T \underline{x}^j$$
$$\textit{subject to}: \sum_{i=1}^{N} \lambda_i z^i = 0 \textbf{ and } \lambda_i \geq 0$$

In the solution, **those points for which λ_i >0 are called support vectors** (primal constraints are active). **Support vectors are critical elements of the training set**. **They lie closest to the decision boundary**!!

Can transform $\underline{x}$ into K($\underline{x}$): Gaussian RBF, Polynomial, MLP, etc. are used as Kernels. Kernels exploit inner products between data points

$$\Rightarrow \textit{replace} \left(\underline{x}^i \right)^T \underline{x}^j \textit{ by } K(\underline{x}^i, \underline{x}^j) \textit{ in the dual.}$$
$$Ex: K(\underline{x}^i, \underline{x}^j) = e^{-\| \underline{x}^i - \underline{x}^j \|_2^2 / 2\sigma^2}; (\left(\underline{x}^i \right)^T \underline{x}^j + 1)^d; \tanh(\gamma \left(\underline{x}^i \right)^T \underline{x}^j + r)$$
$$\textit{They all satisfy } K = [K(\underline{x}^i, \underline{x}^j)] \geq 0 \ (\textit{Mercer's Theorem})$$

Discuss examples of Kernels

C-SVM: $$\min_{\underline{w}} \quad \frac{1}{2} \underline{w}^T \underline{w} + C \sum_{i=1}^{N} \alpha_i$$
$$z^i (\underline{w}^T \underline{x}^i - w_0) - 1 + \alpha_i \geq 0 \quad \textit{and } \alpha_i \geq 0 \quad \forall i$$

$$\max_{\underline{\lambda}} q(\underline{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j z^i z^j \underline{x}^{iT} \underline{x}^j$$
Dual: $$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \| V \underline{\lambda} \|_2^2; V = [z^1 \underline{x}^1, z^2 \underline{x}^2, \ldots, z^N \underline{x}^N]$$
$$\textit{subject to}: \sum_{i=1}^{N} \lambda_i z^i = 0 \textbf{ and } 0 \leq \lambda_i \leq C$$

$$\frac{\beta}{2} \| \underline{w} \|_2^2 + C \sum_{i=1}^{N} [1 - z^i (\underline{w}^T \Phi(\underline{x}^i) - w_0)]^+$$ Hinge loss function

β **regularization weight**

- Pegasos algorithm using sub-gradient method

$$QP\ \text{Problem}:\quad \text{Replace parameter } C \text{ by } \nu \in [0,\ 1]$$

$$\nu\text{=lower bound on the number of support vectors}$$

- ν-SVM: $$\min_{\underline{w}, w_0, \underline{\alpha} \geq \underline{0}, \rho \geq 0} \frac{1}{2} \| \underline{w} \|_2^2 - \nu\rho + \frac{1}{N} \sum_{i=1}^{N} \alpha_i$$

$$s.t.\ \ z^i (\underline{w}^T \Phi(\underline{x}^i) - w_0) \geq \rho - \alpha_i$$

$$Dual: q(\underline{\lambda}) = \quad -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j z^i z^j K(\underline{x}^i, \underline{x}^j)$$

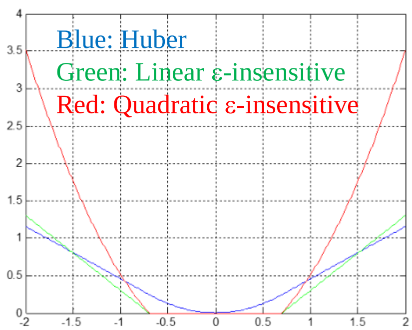
$$\text{subject to}: \sum_{i=1}^{N} \lambda_i z^i = 0 \ \textbf{and}\ 0 \leq \lambda_i \leq \frac{1}{N}; \sum_{i=1}^{N} \lambda_i \geq \nu$$

$$K(\underline{x}^i, \underline{x}^j) = \Phi(\underline{x}^i)^T \Phi(\underline{x}^j)$$

$$\nu \in [0,1] \Rightarrow \text{easy to experiment}$$

$$C = \frac{1}{N\rho}$$

- SVM and elastic net
- SVM Regression



$$e = z - y(\underline{x})$$

$$y(\underline{x}) = \underline{w}^T \underline{\Phi}(\underline{x}) + w_0$$

Huber:

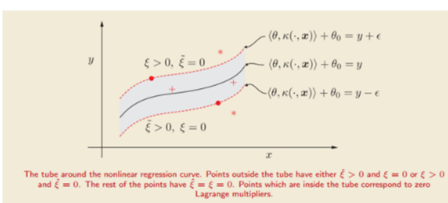
$$L_\varepsilon(e) = \begin{cases} \varepsilon |e| - \dfrac{\varepsilon^2}{2}; |e| > \varepsilon \\ \dfrac{e^2}{2}; |e| \leq \varepsilon \end{cases}$$

Linear ε *– insensitive*:

$$L_\varepsilon(e) = \begin{cases} |e| - \varepsilon; |e| > \varepsilon \\ 0; |e| \leq \varepsilon \end{cases}$$

Quadratic ε *– insensitive*:

$$L_\varepsilon(e) = \begin{cases} e^2 - \varepsilon^2; |e| > \varepsilon \\ 0; |e| \leq \varepsilon \end{cases}$$



The tube around the nonlinear regression curve. Points outside the tube have either $\tilde{\xi} > 0$ and $\xi = 0$ or $\xi > 0$ and $\tilde{\xi} = 0$. The rest of the points have $\tilde{\xi} = \xi = 0$. Points which are inside the tube correspond to zero Lagrange multipliers.