RVMs for classification

Lecture 25

Consider $K=2$: $t \in \{0,1\}$.

$$y(\vec{x},\vec{w}) = \sigma(\vec{w}^T \cdot \vec{\varphi}(\vec{x}))$$

We will use a separate α_i for each w_i.
given $\vec{\alpha} = \underline{\overline{\alpha_1 \ldots \alpha_M}}$, we have:

$$\underbrace{\log p(\vec{w}\,|\,\vec{t},\vec{\alpha})}_{\text{posterior}} = \underbrace{\sum_{n=1}^{N} [t_n \log y_n + (1-t_n)\log(1-y_n)]}_{\log \mathcal{L}} - \qquad (*)$$

$$\underbrace{-\frac{1}{2}\vec{w}^T A \vec{w}}_{\log(\text{prior})} + \text{const}(\vec{w})$$

$$A = \begin{pmatrix} \alpha_1 & & 0 \\ & \ddots & \\ 0 & & \alpha_M \end{pmatrix}$$

Maximize $(*)$ wrt $\vec{w}$ to get $\vec{w}^*$:

$$\frac{\partial}{\partial w_j}[t_n \log y_n + (1-t_n)\log(1-y_n)] =$$

$$= \left[\frac{t_n}{y_n}\underbrace{\varphi_j(\vec{x}_n)}_{y_n(1-y_n)} - \frac{(1-t_n)}{1-y_n}\varphi_j(\vec{x}_n)\right] = \underbrace{\frac{t_n(1-y_n)-(1-t_n)y_n}{y_n(1-y_n)}}_{y_n(1-y_n)}\varphi_j(\vec{x}_n) =$$

$\uparrow$ from $\sigma'(\vec{w}^T\vec{\varphi})$

$$= \varphi_j(\vec{x}_n)(t_n - y_n) = \varphi^T_{jn}(t_n - y_n).$$

So, $\nabla_{\vec{w}} \log p(\vec{w}\,|\,\vec{t},\vec{\alpha})\Big|_{\vec{w}^*} = \varphi^T(\vec{t}-\vec{y}) - A\vec{w}\Big|_{\vec{w}^*} = 0$, so that

$$\vec{w}^* = A^{-1}\varphi^T(\vec{t}-\vec{y}).$$

Furthermore,

$$\nabla_{\vec{w}} \nabla_{\vec{w}} \log p(\vec{w} \mid \vec{t}, \vec{\alpha}) = -\,\varphi^T \underbrace{B}_{N \times N} \varphi - A$$

$$B = \begin{pmatrix} y_1(1-y_1) & & 0 \\ & \ddots & \\ 0 & & y_N(1-y_N) \end{pmatrix}$$

Thus $p(\vec{w} \mid \vec{t}, \vec{\alpha}) \underset{\uparrow}{\simeq} p(\vec{w}^* \mid \vec{t}, \vec{\alpha})\, e^{-\frac{1}{2}(\vec{w}-\vec{w}^*)\Sigma^{-1}(\vec{w}-\vec{w}^*)}$, such that

Laplace appr'n for posterior

$$\Sigma^{-1} = \varphi^T B \varphi + A, \quad \text{evaluated at } \vec{w}^*$$

$$\underbrace{p(\vec{t} \mid \vec{\alpha})}_{\text{evidence for } \vec{\alpha}} = \int d\vec{w}\, p(\vec{t} \mid \vec{w})\, p(\vec{w} \mid \vec{\alpha}) \simeq$$

$$\simeq p(\vec{t} \mid \vec{w}^*)\, p(\vec{w}^* \mid \vec{\alpha})\, (2\pi)^{M/2} |\Sigma|^{1/2}.$$

Now we have

$$\frac{\partial}{\partial \alpha_i} \log p(\vec{t} \mid \vec{\alpha}) = \frac{\partial}{\partial \alpha_i} \log p(\vec{w}^* \mid \vec{\alpha}) + \frac{1}{2} \frac{\partial}{\partial \alpha_i} \log |\Sigma| =$$

$$= \frac{\partial}{\partial \alpha_i}\Big[\frac{1}{2} \underbrace{\log|A|}_{\sum_{j=1}^{M} \log \alpha_j} - \frac{1}{2} \vec{w}^{*T} \underbrace{A}_{\frac{\partial A}{\partial \alpha_i} = \mathbb{1}_{(i,i)}} \vec{w}^*\Big] - \frac{1}{2} \frac{\partial}{\partial \alpha_i} \log |\underbrace{\Sigma^{-1}}_{\varphi^T B \varphi + A}| =$$

$$= \frac{1}{2\alpha_i} - \frac{1}{2}(w_i^*)^2 - \frac{1}{2} \mathrm{Tr}\Big(\Sigma \underbrace{\frac{\partial \Sigma^{-1}}{\partial \alpha_i}}_{\mathbb{1}_{(i,i)}}\Big) =$$

$$= \frac{1}{2\alpha_i} - \frac{1}{2}(w_i^*)^2 - \frac{\Sigma_{ii}}{2} = 0.$$

Then $$\alpha_i (w_i^*)^2 = \underbrace{1 - \alpha_i \Sigma_{ii}}_{\equiv\, \gamma_i}, \text{ or}$$

$$\alpha_i^{new} = \frac{\gamma_i}{\underbrace{(w_i^*)^2}_{\text{evaluated at } \vec{\alpha}^{old},\, \vec{w}^*}} \quad \text{update eq'n}$$

Iterate to convergence between estimating $\left\{ \begin{matrix} \vec{w}^* \\ \Sigma \end{matrix} \right\}$ & $\vec{\alpha}$ $\Rightarrow$ obtain sparse solution, usually much sparser than SVMs

---

We can also analyze α_i dependence more explicitly by using $\log p(\vec{t}\,|\,\vec{\alpha}) \equiv L(\vec{\alpha})$ & splitting off the α_i-dependent terms: $L(\vec{\alpha}) = L(\vec{\alpha}_{-i}) + \lambda(\alpha_i)$ as before

---

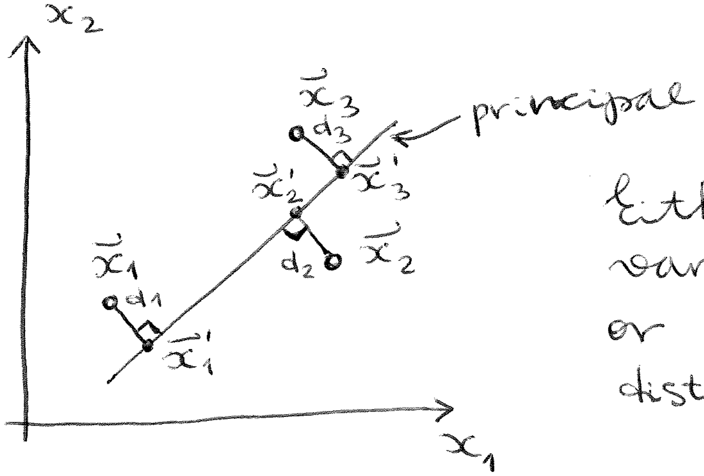
Finally, RVMs generalize to $K > 2$ w/out difficulties:

$$\underbrace{y_k(\vec{x})}_{\text{softmax f'n}} = \frac{e^{a_k}}{\sum_{j=1}^{K} e^{a_j}}, \quad a_k = \vec{w}_k^T \vec{\phi}(\vec{x})$$

Principal component analysis (PCA)

Goals: dimensionality reduction, feature extraction, data visualization

(D=2)



Either maximize the variance of $\vec{x}_1', \vec{x}_2', \vec{x}_3', \ldots$ or minimize the $\perp$ distances $d_1, d_2, d_3, \ldots$

Max variance formulation

Consider $\{\vec{x}_n\}_{n=1}^{N}$, dimensionality $\underbrace{D}_{\#\text{ components in } \vec{x}_n}$

Goal: project $\vec{x}_n$'s onto a subspace with $M < D$, while maximizing the variance of the projected data.

Start with $\underline{M=1}$: define $\underbrace{\vec{u}_1}_{D \text{ dim's}}$ s.t. $\vec{u}_1^T \cdot \vec{u}_1 = 1$

Then $\vec{x}_n \Rightarrow x_n' = \vec{u}_1^T \cdot \vec{x}_n$, and

$$\underbrace{\langle x' \rangle}_{\text{mean}} = \frac{1}{N}\sum_{n=1}^{N} x_n' = \vec{u}_1^T \cdot \underbrace{\frac{1}{N}\sum_n \vec{x}_n}_{\langle \vec{x} \rangle} = \vec{u}_1^T \cdot \langle \vec{x} \rangle$$

$$\mathrm{Var}\{x_n'\} = \frac{1}{N}\sum_n \left(\vec{u}_1^T \cdot \vec{x}_n - \vec{u}_1^T \cdot \langle \vec{x} \rangle\right)^2 =$$

$$= \frac{1}{N}\sum_n u_{1,k}\left[x_{n,k}x_{n,j} + \langle x \rangle_k \langle x \rangle_j - x_{n,k}\langle x \rangle_j - x_{n,j}\langle x \rangle_k\right] u_{1,j} =$$

$$\stackrel{\textcircled{=}}{} u_{1,k}\Big\{\underbrace{\frac{1}{N}\sum_n (x_{n,k}-\langle x\rangle_k)(x_{n,j}-\langle x\rangle_j)}_{\overset{\text{III}}{S_{kj}}}\Big\} u_{1,j} =$$

$$= \underbrace{\vec{u}_1^T S \vec{u}_1}_{\text{maximize}}$$ this, subject to $\vec{u}^T \cdot \vec{u} = 1$:

$$\mathcal{J} = \vec{u}_1^T S \vec{u}_1 + \underset{\uparrow}{\lambda_1}(1 - \vec{u}_1^T \vec{u}_1).$$

Lagrange multiplier

$$\frac{\partial \mathcal{J}}{\partial \vec{u}_1} = 2(S\vec{u}_1 - \lambda_1 \vec{u}_1) = 0 \quad \Rightarrow \quad \underline{\underline{S\vec{u}_1 = \lambda_1 \vec{u}_1}}$$

Thus $\vec{u}_1$ is an eigenvector of S, with eigenvalue λ_1.

Note that $\vec{u}_1^T S \vec{u}_1 = \lambda_1$,
the variance will be at max if λ_1
is the <u>largest</u> eigenvalue of S; corresponding
$\vec{u}_1$ is called the <u>1st principal component</u>.

We can continue to project, s.t. for M-dim
principal subspace the optimal projection
is defined by M largest eigenvalues:

$\lambda_1, \ldots, \lambda_M$

$\vec{u}_1, \ldots, \vec{u}_M \Leftarrow$ corresponding eigenvectors

<u>Proof by induction</u>:

$M=1$ shown above.

For $M > 1$, assume this holds for M & prove it for $M+1$. Thus

$\vec{u}_1, \dots, \vec{u}_M$ are principal eigenvectors

and we are looking for $\vec{u}_{M+1} \perp \vec{u}_1, \dots,$

$\vec{u}_{M+1} \perp \vec{u}_M$.

$$\vec{u}^T_{M+1} \cdot \vec{u}_{M+1} = 1$$

otherwise $\vec{u}_{M+1}$ lies in the M-subspace

Since Var in the $\vec{u}_{M+1}$ direction is given by $\vec{u}^T_{M+1} S \vec{u}_{M+1}$, we need to maximize

$$\mathcal{L} = \vec{u}^T_{M+1} S \vec{u}_{M+1} + \lambda_{M+1}(1 - \vec{u}^T_{M+1} \cdot \vec{u}_{M+1}) +$$
$$+ \sum_{i=1}^{M} \eta_i \vec{u}^T_{M+1} \cdot \vec{u}_i$$

$$\frac{\partial \mathcal{L}}{\partial \vec{u}_{M+1}} = 0 \Rightarrow 2S\vec{u}_{M+1} - 2\lambda_{M+1}\vec{u}_{M+1} + \sum_{i=1}^{M} \eta_i \vec{u}_i = 0.$$

$$2\vec{u}^T_{M+1} S \vec{u}_{M+1} - 2\lambda_{M+1} \underbrace{\vec{u}^T_{M+1} \vec{u}_{M+1}}_{1} + \sum_{i=1}^{M} \eta_i (\underbrace{\vec{u}^T_{M+1} \vec{u}_i}_{0}) = 0,$$

or

$$\vec{u}^T_{M+1} S \vec{u}_{M+1} = \lambda_{M+1} \Leftarrow \text{maximized when } \lambda_{M+1} \text{ is the largest remaining eigenvalue}$$

$$\Downarrow$$

$$S\vec{u}_{M+1} = \lambda_{M+1} \vec{u}_{M+1}.$$

Min-error formulation

Again, introduce $\{\vec{u}_i\}_{i=1}^{D}$ s.t.

$$\vec{u}_j^T \cdot \vec{u}_i = \delta_{ij}$$

Then $\vec{x}_n = \sum_{i=1}^{D} \alpha_{ni} \vec{u}_i = \sum_{i=1}^{D} (\vec{x}_n^T \cdot \vec{u}_i) \vec{u}_i$

indep. of n

Approximate $\vec{x}_n \Rightarrow \tilde{\vec{x}}_n = \sum_{i=1}^{M} z_{ni} \vec{u}_i + \sum_{i=M+1}^{D} b_i \vec{u}_i$

Now, choose $\{\vec{u}_i\}$, $\{z_{ni}\}$, $\{b_i\}$ to minimize

$$J = \frac{1}{N} \sum_{n=1}^{N} \| \vec{x}_n - \tilde{\vec{x}}_n \|^2 .$$

$$\frac{\partial J}{\partial z_{nj}} = \frac{2}{N} \left(\vec{x}_n - \sum_{i=1}^{M} z_{ni} \vec{u}_i - \sum_{i=M+1}^{D} b_i \vec{u}_i \right)^T \cdot \vec{u}_j \underset{1 \le j \le M}{=}$$

$$= \frac{2}{N} \left(\vec{x}_n^T \cdot \vec{u}_j - z_{nj} \right) = 0 \Rightarrow z_{nj} = \vec{x}_n^T \cdot \vec{u}_j .$$

Likewise,

$$\frac{\partial J}{\partial b_i} = \frac{2}{N} \sum_{n=1}^{N} \left(\vec{x}_n - \sum_{i=1}^{M} z_{ni} \vec{u}_i - \sum_{i=M+1}^{D} b_i \vec{u}_i \right)^T \cdot \vec{u}_j \underset{M+1 \le j \le D}{=}$$

$$= 2 \left(\frac{1}{N} \sum_{n=1}^{N} (\vec{x}_n^T \cdot \vec{u}_j) - b_j \right) = 0 , \text{ or}$$

$$b_j = \langle \vec{x} \rangle^T \cdot \vec{u}_j .$$

Now,

$$\vec{x}_n - \tilde{\vec{x}}_n = \underbrace{\sum_{i=M+1}^{D} \left[(\vec{x}_n - \langle \vec{x} \rangle)^T \cdot \vec{u}_i \right] \vec{u}_i}$$

"residual" lies in the subspace defined by $\vec{u}_{M+1} \dots \vec{u}_D$

Furthermore,

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (\vec{x}_n^T \cdot \vec{u}_i - \langle \vec{x} \rangle^T \cdot \vec{u}_i)^2 =$$

$$= \sum_{i=M+1}^{D} \vec{u}_i^T S \vec{u}_i \quad .$$

Consider $D=2$, $M=1$ case first:

minimize $J = \vec{u}_2^T S \vec{u}_2$ subject to $\vec{u}_2^T \cdot \vec{u}_2 = 1$.

$$\tilde{J} = \vec{u}_2^T S \vec{u}_2 + \lambda_2 (1 - \vec{u}_2^T \vec{u}_2).$$

$$\frac{\partial \tilde{J}}{\partial \vec{u}_2} = 0 \Rightarrow \underline{S\vec{u}_2 = \lambda_2 \vec{u}_2} \Rightarrow J = \lambda_2$$

To minimize J, choose λ_2 to be the smaller eigenvalue $\Rightarrow$ the principal subspace is defined by $\vec{u}_1$ corresponding to λ_1, the larger eigenvalue.

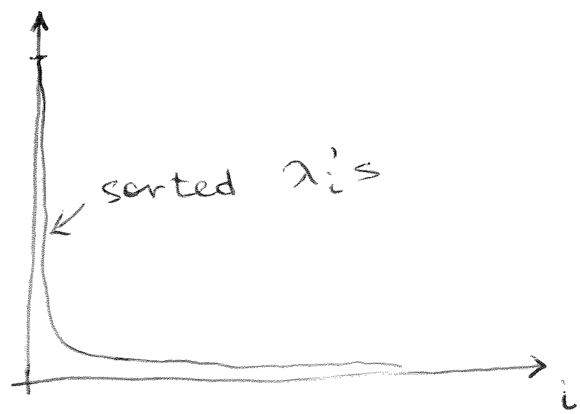
In general, for $M<D$ solve

proof left as an exercise

$$S\vec{u}_i = \lambda_i \vec{u}_i \qquad i=1,\dots,D$$

Then $J=\sum_{i=M+1}^{D}\lambda_i$ is minimized by choosing $\lambda_{M+1},\dots,\lambda_D$ to be the smallest eigenvalues $\Rightarrow$ $\lambda_1,\dots,\lambda_M$ are the largest eigenvalues & $\vec{u}_1,\dots,\vec{u}_M$ define the principal subspace.

Typically, we have for high-D data:



clearly, $J\searrow$ as $M\nearrow$

Now, $$\tilde{\vec{x}}_n = \sum_{i=1}^{M}(\vec{x}_n^T\cdot\vec{u}_i)\vec{u}_i + \sum_{i=M+1}^{D}(\langle\vec{x}\rangle^T\cdot\vec{u}_i)\vec{u}_i \;=$$

$$\langle\vec{x}\rangle = \sum_{i=1}^{D}(\langle\vec{x}\rangle^T\cdot\vec{u}_i)\vec{u}_i \quad \text{gives}$$

$$= \langle\vec{x}\rangle + \sum_{i=1}^{M}\underbrace{(\vec{x}_n^T\cdot\vec{u}_i - \langle\vec{x}\rangle^T\cdot\vec{u}_i)}_{\text{components of an M-dim vector}}\vec{u}_i$$

This amounts to <u>data compression</u>

We can also use PCA to standardize the data.

Write $S\vec{u}_i = \lambda_i \vec{u}_i$ as $S\mathcal{U} = \mathcal{U}L$

each vector is a column

$$\mathcal{U} = \begin{pmatrix} \vec{u}_1 & \dots & \vec{u}_D \end{pmatrix} \qquad L = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix}$$

$\mathcal{U}^T = \mathcal{U}^{-1}$

Define $\vec{y}_n = L^{-1/2}\,\mathcal{U}^T(\vec{x}_n - \langle \vec{x} \rangle)$.

Note that $\langle \vec{y} \rangle = \frac{1}{N}\sum_{n=1}^{N} \vec{y}_n = 0$, and covariance is then given by

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N} \vec{y}_n \cdot \vec{y}_n^{\,T}}_{D \times D \text{ matrix}} = \frac{1}{N}\sum_{n=1}^{N} L^{-1/2}\,\mathcal{U}^T(\vec{x}_n - \langle \vec{x} \rangle)(\vec{x}_n - \langle \vec{x} \rangle)^T * \mathcal{U}L^{-1/2} =$$

$$= L^{-1/2}\,\underbrace{\mathcal{U}^T S\,\mathcal{U}}_{\mathcal{U}^{-1}\mathcal{U}L = L}\,L^{-1/2} = \mathbb{1}.$$

This operation is known as <u>whitening</u> the data.

Finally, $M=2$ projections can be used for data <u>visualization</u>.

PCA for high-D data

Sometimes, $N < D$ $\Rightarrow$ a set of N points occupy a subspace with at most $N-1$ dims, so using $M > N-1$ is pointless since at least $D-N+1$ λ_i's are $\emptyset$. Doing "typical" PCA (i.e. finding all λ_i's) scales as $\Theta(D^3)$ and is too costly.

To address this problem, define

$$\underbrace{X}_{N\times D} = \begin{pmatrix} \vec{x}_1 - \langle \vec{x} \rangle \\ \vdots \\ \vec{x}_N - \langle \vec{x} \rangle \end{pmatrix}$$

Then $\overset{D\times D}{\tilde{S}} = \frac{1}{N} X^T X$, and

$$\underbrace{\frac{1}{N} X^T X}_{S} \vec{u}_i = \lambda_i \vec{u}_i$$

$$\Downarrow$$

eigenvector eq'n $\Rightarrow$ $$\underbrace{N^{-1} X X^T}_{N\times N} \underbrace{(X \vec{u}_i)}_{\equiv \vec{v}_i} = \lambda_i (X \vec{u}_i) \qquad (*)$$

, assume $\vec{v}_i^T \cdot \vec{v}_i = 1$ w/out loss of generality

λ_i, $i = 1 \ldots N$ are the same as the "original" λ_i's, and the other λ_i's are all $\emptyset$.

Solving $(*)$ is an $\Theta(N^3)$ operation.

Finally, eigenvectors:

$$\underbrace{\left(N^{-1} X^T X\right)}_{S} \underbrace{(X^T \vec{v}_i)}_{\substack{\text{eigenvector} \\ \text{of } S \text{, length } D}} = \lambda_i (X^T \vec{v}_i)$$

Normalization: $(X^T \vec{v}_i)^T (X^T \vec{v}_i) =$

$$= \vec{v}_i^T \underbrace{X X^T \vec{v}_i}_{N \lambda_i \vec{v}_i} = N \lambda_i \vec{v}_i^T \vec{v}_i$$

Assuming that $\vec{v}_i^T \vec{v}_i = 1$, we obtain: $\dfrac{X^T \vec{v}_i}{\sqrt{N \lambda_i}}$ as normalized eigenvectors of S

Thus, we construct $N^{-1} X X^T$, compute its eigenvalues λ_i & $\underbrace{\text{eigenvectors}}_{\text{normalized}}$ $\underbrace{\vec{v}_i}_{\text{length } N}$,

and then project back into D-space using $\underbrace{\dfrac{X^T \vec{v}_i}{\sqrt{N \lambda_i}}}_{\text{length } D}$. ← normalized eigenvectors in original data space