SML

Signal Analysis, Models, and Machine Learning

A Summary

Part I Linear Signal Representation and Approximation

Chapter 1: Hilbert Spaces

A Hilbert space is a vector space with some additional properties.

1.1 Vector Spaces

Is a generalization of the familiar space $\mathbb{R}^n$ of real n-tuples.

Def 1.1 Let $\mathcal{F} \triangleq \mathbb{R}$ or $\mathcal{F} \triangleq \mathbb{C}$. A vector space over $\mathcal{F}$ is a set V with properties:

1. Operation "+" : $V \times V \to V$
   a) $(u+v)+w = u+(v+w)$ associative law
   b) $v+w = w+v$ commutative law
2. There exists $0_V \in V$ s.t. $0_V + v = v \quad \forall v \in V$
3. There exists $(-v) \in V$ s.t. $v + (-v) = 0_V \quad \forall v \in V$
4. Operation "multiplication" $\mathcal{F} \times V \to V$
   a) $a(v+w) = av + aw$ distributive laws
   b) $(a+b)v = av + bv$
   c) $(ab)v = a(bv)$
   d) $1 \cdot v = v$

Elements of V are called vectors, elements of $\mathcal{F}$ are scalars.

Thm 1.1 For any $v \in V$ and $a \in \mathcal{F}$ the following are true

1. There is only one zero element 0_V in V
2. There is only one $v' \in V$ s.t. $v + v' = 0_V$
3. $0 \cdot v = a \cdot 0_V = 0_V$
4. $(-a)v = a(-v) = -(av)$

A subspace of a vector space V is a subset of V that is a vector space in its own right (with same operations as V and same field $\mathcal{F}$).

Thm 1.2 Subspace Test A nonempty subset U of some vector space V over $\mathcal{F}$ is a subspace of V iff the following conditions are satisfied:

1. $au \in U \quad \forall a \in \mathcal{F}, \forall u \in U$
2. $u + u' \in U \quad \forall u, u' \in U$

Proof: Take an U and check all props. defined in Def 1.1

Thm 1.3 Intersection Thm for Vector Spaces If $U_1, U_2, \ldots$ are subspaces of V, then $U_1 \cap U_2 \cap \ldots$ is also a subspace of V.

1.2 Linear Independence and Basis

Let $S = \{v_1, v_2, \dots\} \subset V$ be a countable subset of some vectorspace V over $\mathcal{F}$. The subspace spanned by S is the set:

$$\operatorname{span}(v_1, v_2, \dots) \triangleq \left\{ \sum_k a_k v_k : a_k \in \mathcal{F} \right\}$$

(all linear combinations of the vectors in S)

Thm 1.4 $\operatorname{span}(v_1, v_2, \dots)$ is a subspace of V

A vector space V is called finite-dimensional if $V = \operatorname{span}(S)$ for some finite set $S \subset V$. The vectors $v_1, v_2, \dots \in V$ are called *linearly independent* if

$$a_1 v_1 + a_2 v_2 + \dots = 0_V$$

with $a_i \in \mathcal{F}$ has only the solution $a_1 = a_2 = \dots = 0$. Linear dependent if ↖ has another solution.

Thm 1.5 Unique-Representation Thm Let V be some vector space over $\mathcal{F}$ and let $v_1, \dots \in V$ be lin. indep. If $v \in V$ can be written as

$$v = \sum_k a_k v_k = \sum_k \beta_k v_k$$

for some $a_k, \beta_k \in \mathcal{F}$, then $\boxed{a_k = \beta_k}$ for $k = 1, \dots, n$

A basis of V over $\mathcal{F}$ is a countable set $\{v_1, \dots\} \subset V$ of lin. indep. vectors, s.t. $V = \operatorname{span}(v_1, \dots)$
Eg $\{(1,0,0), (0,1,0), (0,0,1)\}$ is the basis of the vector space $\mathcal{F}^3$.

Thm 1.6 Basis Extention Thm Any set of lin. indep. vectors $\{v_1, v_2, \dots, v_m\} \subset V$ can be extended to a basis of V

Thm 1.7 If both $v_1, v_2, \dots v_m$ and $w_1, w_2, \dots, w_n$ form a basis of V, then $n = m$

$\dim V$ is the dimension of V, the unique number of elements in any basis of V

1.3 Inner Product and Norm

notation: complex conjugate: $\overline{x}$, $A^H \triangleq \overline{A^T}$

Def 1.2 V a vector space over $\mathcal{F}$. Inner product is a function $V \times V \to \mathcal{F} : (v, w) \mapsto \langle v, w \rangle$

1. $\langle u+v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$
2. $\langle v, w \rangle = \overline{\langle w, v \rangle}$
3. $\langle av, w \rangle = a \langle v, w \rangle$
4. $\langle v, v \rangle \geq 0$
4.a $\langle v, v \rangle = 0 \Leftrightarrow v = 0_V$
$\|v\| \triangleq \sqrt{\langle v, v \rangle}$
5. $\langle u, v+w \rangle = \langle u, v \rangle + \langle u, w \rangle$
6. $\langle v, aw \rangle = \bar{a} \langle v, w \rangle$
7. $\langle v, 0_V \rangle = 0$

Some examples: In $\mathcal{F}^n$ $\langle v, w \rangle \triangleq \sum v_k \overline{w_k} = w^H v$ generally $\langle v, w \rangle \triangleq \sum a_k v_k \overline{w_k} = w^H A v$
$A \triangleq \operatorname{diag}(a_1, \dots, a_n)$

In $\mathbb{Z} \to \mathcal{F}$ $\langle f, g \rangle \triangleq \sum f[k] \overline{g[k]}$ $T \to \mathcal{F}, T = \mathbb{R}$ $\langle f, g \rangle = \int_T f(t) \overline{g(t)} dt$

Thm 1.8 Schwarz Inequality $|\langle v, w \rangle| \leq \|v\| \cdot \|w\|$ for all $v, w \in V$

Thm 1.9 1. $\|v\| \geq 0$ 2. $\|v+w\| \leq \|v\| + \|w\|$ 3. $\|av\| = |a| \cdot \|v\|$ → $\|v-w\| \leq \|v\| + \|w\|$
$\|v\| = 0 \Leftrightarrow v = 0_V$

1.4 Technicalities

The inner product $\langle f,g\rangle = \sum f[k]\overline{g[k]}$ may fail to converge. Or in $\langle f,g\rangle = \int_T f(t)\overline{g(t)}dt$ if $\|f\| = \int_T |f(t)|^2 dt = 0$ we cannot conclude that $f(x)=0$ for all $x \in T$. This problem is solved by the following trick: For any $v \in V$, let $[v] \triangleq \{v' \in V : \|v-v'\| = 0\}$ is an equivalence class. The vector space V is then replaced by the space $\{[v] : v \in V\}$.

→ Have to redefine equality: "$f=g$" means $\int_T |f(t)-g(t)|^2 dt = 0$

"$X=Y$" means $E[|X-Y|^2] = 0$ equiv. $P(X=Y)=1$

"$X=Y$" with probability one (w. p. 1)

Thm 1.10 Let $\mathcal{F}=\mathbb{R}$ or $\mathcal{F}=\mathbb{C}$ and V_{fin} set of all functions $f: \mathbb{Z} \to \mathcal{F}$ that satisfies $\sum_{k\in\mathbb{Z}} |f[k]|^2 < \infty$

1. V_{fin} is a vector space (a subspace of V)
2. The inner product $\sum_{k\in\mathbb{Z}} f[k]\overline{g[k]}$ is well defined and finite $\forall f,g \in V_{fin}$

Thm 1.11 V: space of all $\mathcal{F}$-valued r.v. $V_{fin} \subset V$ set of those r.v. X for which both $E[X]$ and $E[|X|^2]$ are well defined and finite

1. V_{fin} is a vector space (subspace of V)
2. Inner product (correlation) $E[X\overline{Y}]$ is well defined and finite $\forall X,Y \in V_{fin}$

1.4.3 Hilbert Spaces

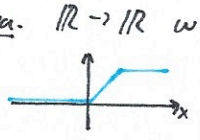
A vector space with an inner product is a Hilbert space iff an additional cond. is met:

Def 1.3 A vector space V with an inner product is a Hilbert space if every Cauchy sequence of elements in V converges to an element of V.

Cauchy sequence: $v_1, v_2, \ldots \in V$. for every $\varepsilon > 0$ there exists a positive integer n_0 s.t. $\|v_k - v_m\| < \varepsilon$ holds for all $m > n_0$, $k > n_0$.

Ex 1.11 Set of cont. fcn. $\mathbb{R} \to \mathbb{R}$ with inner product, finite norm is a vector space. But

$$f_n(x) = \begin{cases} 0 & x \le 0 \\ nx & 0 \le x \le 1/n \\ 1 & x \ge 1/n \end{cases}$$



for $n = 1, 2, \ldots$ form a Cauchy seq. of cont. fcn. which converges to the step function  which is not cont. and not in $V \to$ no Hilbert space.

Some Hilbert Spaces: • $\mathcal{F}^n$ • The set $e_{\mathcal{F}}^2(\mathbb{Z})$ of functions $\mathbb{Z} \to \mathcal{F}$ (discrete time signals) with fin. energy

• $L_{\mathcal{F}}^2(\mathbb{R})$ square integrable (finite energy) fnctions $\mathbb{R} \to \mathcal{F}$ • $L_{\mathcal{F}}^2(T)$, T an interval of $\mathbb{R}$

• Set of $\mathcal{F}$-valued random variables with finite first and second moments (mean & variance)

1.5 Orthogonal Vectors and Orthonormal Bases

Two vectors in a HS (Hilbert Space) are orthogonal if $\langle v,w\rangle = 0$. We then have

$$\|v+w\|^2 = \|v\|^2 + \|w\|^2 \qquad \|v-w\|^2 = \|v\|^2 + \|w\|^2 \leftarrow \text{Pythagoras' Theorem}$$

For random variables: $E[|X+Y|^2] = E[|X|^2] + E[|Y|^2]$

Thm 1.13 Let $v_1, v_2, \dots$ be pairwise orthogonal (ie $\langle v_n, v_m\rangle = 0$ if $n \neq m$) and $v_n \neq 0_V$ $n = 1,2,\dots$. Then $v_1, v_2, \dots$ are linearly independent.

A subset B of a HS is orthonormal if $\langle v,w\rangle = \begin{cases} 1, & v = w \\ 0, & v \neq w \end{cases}$ for all $v,w \in B$. An orthonormal basis is a basis of orthonormal vectors.

Thm 1.14 Orthonormal-Representation Thm Let V be a HS with orthonormal basis B. Then

$$v = \sum_{w \in B} \langle v,w\rangle w \qquad \text{for every } v \in V$$

Thm 1.15 Parseval's Thm Let V be a HS with orthonormal basis B: $\langle u,v\rangle = \sum_{w \in B} \langle u,w\rangle \overline{\langle v,w\rangle}$ $\forall u,v \in V$

Thm 1.16 $u_1, u_2, \dots$ orthonormal vecs in HS V, $U \triangleq \text{span}(u_1, u_2, \dots)$. $x \in V$ then $v \triangleq x - \sum_k \langle x, u_k\rangle u_k$ is orthogonal to every $u \in U$

Thm 1.17 Gram-Schmidt Orthonormalisation Let V be a vector space and y_i lin. indep. in V. Let u_i be defined as $u_k \triangleq \frac{\tilde{u}_k}{\|\tilde{u}_k\|}$ with $\tilde{u}_1 \triangleq y_1$ and $\tilde{u}_k = y_k - \sum_{i=1}^{k-1} \langle y_k, u_i\rangle u_i$ for $k > 1$ then the set $\{u_1, \dots, u_n\}$ is an orthonormal basis of $\text{span}(y_1, \dots, y_n)$ for $n = 1,2,\dots$.

Ex (Prob 2.3) Find an orthonormal basis for the subspace U of $\mathbb{R}^3$ spanned by $y_1 = (1,\sqrt{2},1)^T$ $y_2 = (1,0,1)^T$

$u_1 = \frac{\tilde{u}_1}{\|\tilde{u}_1\|} = \frac{y_1}{\|y_1\|} = (1/2, 1/\sqrt{2}, 1/2)^T$ $\tilde{u}_2 = y_2 - \langle y_2, u_1\rangle u_1 = (1/2, -1/\sqrt{2}, 1/2)^T$ $u_2 = \frac{\tilde{u}_2}{\|\tilde{u}_2\|} = (1/2, -1/\sqrt{2}, 1/2)^T$

1.6 Orthogonal Complement

Let $\mathbb{F} = \mathbb{R}/\mathbb{F} = \mathbb{C}$ and U be a subspace of some HS V over $\mathbb{F}$. The orthogonal complement $U^\perp$ of U is the set $U^\perp \triangleq \{v \in V : \langle v,u\rangle = 0 \text{ for all } u \in U\}$

Thm 1.18 The orthogonal complement $U^\perp$ is a subspace of V and $U \cap U^\perp = \{0_V\}$.

Thm 1.19 Let $U \triangleq \text{span}(y_1, \dots) \subset V$. Then for any $v \in V$ we have $v \in U^\perp \Leftrightarrow \langle v, y_k\rangle = 0$, $k = 1,2,\dots$

Thm 1.20 Orthogonal decomposition Thm Let V be a HS and U a subspace of V spanned by $\{y_1, \dots\} \subset V$. Then every $x \in V$ has a unique decomposition into $x = u + v$ $u \in U$ $v \in U^\perp$ implies $\dim U + \dim U^\perp = \dim V$

1.7 Projection to a Subspace

Let U a subspace of HS V. The projection of $x \in V$ to U is the unique vector u in the decomposition $x = u+v$, $u \in U$ $v \in U^\perp$.

Thm 1.21 The projection $V \to U : x \mapsto \hat{x}$ is a linear mapping, ie, $ax \mapsto a\hat{x}$ and $x_1 + x_2 \mapsto \hat{x}_1 + \hat{x}_2$

Thm 1.22 Orthogonality Principle Let $U \triangleq \text{span}(y_1, y_2, \dots)$ a subspace of some HS V. Then $u \in U$ is the projection of $x \in V$ to U if and only if $\langle x - u, y_k\rangle = 0$, $k = 1,2,\dots$

Thm 1.23 Orthonormal-Projection Thm V a HS, U subspace of V with orthonormal basis $\{u_1, u_2, \dots\}$. Then the projection of $x \in V$ to U is $u = \sum_k \langle x, u_k\rangle u_k$

Thm 1.24 Orthonormal Projection in lin. Alg. For $\mathbb{F} = \mathbb{R}/\mathbb{F} = \mathbb{C}$ let $B = (b_1, b_2, \dots) \in \mathbb{F}^{m \times n}$ $\begin{bmatrix} + & n & - \\ m & & \\ | & & \end{bmatrix}$ with orthonormal columns $b_1, \dots, b_n \in \mathbb{F}^m$ $b_k^H b_\ell = \begin{cases} 1 & k = \ell \\ 0 & k \neq \ell \end{cases}$ Then the projection of $x \in \mathbb{F}^m$ to the space spanned by $b_1, b_2, \dots, b_n$ is $u = BB^H x$

Proof: $u = \sum b_k \langle x, b_k\rangle = \sum b_k b_k^H x = \underbrace{(b_1, \dots b_n)}_{B} \begin{pmatrix} b_1^H x \\ \vdots \\ b_n^H x \end{pmatrix} = BB^H x$

A Basic concepts of Probability Theory

Probability space $(\Omega, \mathcal{E}, P)$: Ω: experimental outcomes; $\mathcal{E}$: set of events ($=$ subset of Ω) closed under unions and intersections; $P: \mathcal{E} \to [0,1]$ probability measure

Discrete random variable: is a function $X: \Omega \to S$ for every $s \in S$, $\{\omega \in \Omega : X(\omega) = s\}$ is an event. By "the event $X=s$" we mean the set (event) $\{\omega \in \Omega : X(\omega) = s\}$.

Probability mass function: For some discrete r.v. X with codomain S is the function $S \to [0,1] : s \mapsto P(X=s)$ $p_X(\cdot) = p(x)$

$p_X(x) \triangleq P(X=x)$ $\quad p_{X,Y}(x,y) \triangleq P(X=x \text{ and } Y=y)$ $\quad p_{X|A}(x) \triangleq P(X=x|A)$ $\quad p_{X|Y}(x|y) \triangleq P(X=x|Y=y)$

$\sum_y p(x,y) = p(x)$ $\quad$ **Independence**: If $P(X=x, Y=y) = P(X=x) \cdot P(Y=y)$

Conditional probability: $P(A|B) = \dfrac{P(A \cap B)}{P(B)}$ $\quad P(A) = \sum_i P(A|B_i) \cdot P(B_i)$ $\quad P(B_i|A) = \dfrac{P(A|B_i) \cdot P(B_i)}{P(A)}$

Distribution function: $F_X: \mathbb{R} \to \mathbb{R}$ $\quad F_X(r) = P(X \le r)$ $\quad$ **density**: $f_X(x) \triangleq \dfrac{d}{dx} F_X(x)$ $\quad \int_{-\infty}^{\infty} f(x,y)\,dy = p(x)$

$p(x,y,z) = p(x)p(y|x)p(z|x,y)$ $\quad E[X] = \sum_y E[X|Y=y] \cdot p(y)$ $\quad P(A|Y=y) = \dfrac{P(A)\,p_{Y|A}(y)}{p_Y(y)}$ $\quad P(A) = \sum_y p_Y(y) P(A|Y=y)$

Mean (Expectation): $m_X = E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x)\,dx$ $\quad E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\,dx$ $\quad E[aX+bY] = a E[X] + b E[Y]$

X, Y indep. $\Rightarrow$ X, Y uncorrelated $\Leftrightarrow$ $E[X \cdot \overline{Y}] = E[X] \cdot \overline{E[Y]}$

Variance: $\mathrm{Var}(X) \triangleq E[(X-m_X)^2] = E[X^2] - m_X^2$ $\quad$ if X complex: $\mathrm{Var}(X) \triangleq E[(X-m_X)\overline{(X-m_X)}] = E[|X|^2] - |m_X|^2$

X, Y uncorrelated: $\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$ $\quad$ If X, Y orthogonal ($E[X \cdot \overline{Y}] = 0$): E

Correlation matrix: $R_X \triangleq E[XX^H]$ $\quad$ **Cov. matrix**: $V_X \triangleq E[(X-m_X)(X-m_X)^H] = E[XX^H] - m_X m_X^H$

$R_{AX} = A R_X A^H$ $\quad V_{AX} = A V_X A^H$

Weak law of large numbers: If $X_1, X_2, \ldots$ are pairwise uncorrelated and identically distributed with finite mean $E[X_k] = m_X$ and finite variance, then

$$\lim_{n \to \infty} P\left(\left| m_X - \frac{1}{n} \sum_{k=1}^{n} X_k \right| < \varepsilon \right) = 1 \quad \text{for any positive } \varepsilon.$$

Strong law of large numbers: If $X_1, \ldots$ iid with finite mean and $E[|X_k|] < \infty$

$$\lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} X_k = m_X \quad \text{with probability one.}$$

B Concepts & Terms of estimation theory

ML: Maximum Likelihood Estimation Given: probability law $p(y|x)$ and observation $Y=y$

Estimate: $\hat{x} = \underset{x}{\operatorname{argmax}}\ \underbrace{p(y|x)}_{\text{likelihood function}}$

- Given samples y, find a distribution p with parameters x that fits best.

MAP: maximum a-posteriori Given: joint probability $p(x,y)$ and observation $Y=y$

Estimate $\hat{x} = \underset{x}{\operatorname{argmax}}\ p(x|y) = \underset{x}{\operatorname{argmax}}\ p(x,y) = \underset{x}{\operatorname{argmax}}\ p(y|x)\,p(x)$

Bayesian Estimation Given: joint probability $p(x,y)$, observation $Y=y$ and cost function $u(x,\hat{x})$

Estimate: $\hat{x} = \underset{x'}{\operatorname{argmin}}\ E[u(X,x')|Y=y] = \underset{x'}{\operatorname{argmin}} \sum_x p(x|y)\,u(x,x') = \underset{x'}{\operatorname{argmin}} \sum_x p(x,y)\,u(x,x')$

MMSE: Bayesian Min. Mean squared Error Bayesian Estimation with $u(x,\hat{x}) = \|x-\hat{x}\|^2 \rightarrow \boxed{\hat{x} = E[X|Y=y]}$

Penalized-Likelihood Given: $p(y|x)$, preference function $\rho(x) \geq 0$, observation $Y=y$

Estimate: $\hat{x} = \underset{x}{\operatorname{argmax}}\ p(y|x)\rho(x) = \underset{x}{\operatorname{argmin}} \left(-\log p(y|x) - \log \rho(x)\right)$

Conjugate Priors and Virtual-Observation Priors

A virtual-observation prior is a preference function $\rho(x)$ of the form $\rho(x) = p(y_0|x)$ for some hypothetical initial/additional "observation" $Y_0 = y_0$. A **conjugate prior** is a prior $p(x)$ s.t. the **posterior** $p(x|y) \propto p(y|x)p(x)$ has the same functional form as the likelihood function $p(y|x)$

Non-Bayesian Esti. for Mean, Var, Cov. Mtx Given: $Y_1,\dots,Y_L$ iid rv with values in $\mathbb{R}^n$ and unknown mean μ_Y and cov. matrix V_Y. Let $y^{(1)},\dots,y^{(L)}$ be realizations of Y_i from which we estimate μ_Y and V_Y.

$\hat{\mu}_Y = \frac{1}{L}\sum_{\ell=1}^{L} y^{(\ell)}$ satisfies $E[\hat{\mu}_Y] = \mu_Y$ and $\lim_{L\to\infty} \hat{\mu}_Y = \mu_Y$ wp.1

Sample covariance matrix $V_S \overset{\Delta}{=} \frac{1}{L}\sum_{\ell=1}^{L} (y^{(\ell)}-\hat{\mu})(y^{(\ell)}-\hat{\mu})^T = \frac{1}{L}\sum y^{(\ell)}(y^{(\ell)})^T - \hat{\mu}\hat{\mu}^T$

satisfies $E[V_S] = E[Y_1 Y_1^T] - E[\hat{\mu}\hat{\mu}^T] = \frac{L-1}{L} V_Y$ and $\lim_{L\to\infty} V_S = V_Y$ wp.1

If $Y_1,\dots,Y_L$ are Gaussian, $\hat{V}_Y = V_S$.

For small L gets problematic. Regularizations such as $\hat{V}_Y = V_S + \frac{\rho}{L} I$, $\rho \in \mathbb{R}$ ~~ρ~~ $\rho = \text{const}$ or $\operatorname{tr}(V_S)$

1.8 Approximation in a Subspace, Least Squares, LMMSE

Least Squares Given: • vector $x=(x_1,\dots,x_m)^T$ with entries in $\mathcal{F}=\mathbb{R}$ or $\mathcal{F}=\mathbb{C}$
• matrix $A\in\mathcal{F}^{m\times n}$ over $\mathcal{F}$

Problem: Find $h=(h_1,\dots,h_n)^T$ such that $\boxed{\|Ah-x\|}$ is as small as possible.

LMMSE (Linear Minimum Mean Squared Error)

Problem: Wish to estimate a random variable X from random variables $Y_1, Y_2, \dots$

$$\hat{X}=\sum_n h_n Y_n \quad \text{s.t.} \quad E[|\hat{X}-X|^2] \quad \text{is as small as possible.}$$

1.8.1 HS problem and solution

Let $x, y_1, y_2, \dots$ be vectors in some HS V. Wish to find $\hat{x}=\text{span}(y_1,\dots)$ s.t. $\|\hat{x}-x\|$ is as small as possible.

Thm 1.25 Let $\{y_1,y_2,\dots\}$ be a countable subset of HS V and $U\triangleq\text{span}(y_1,\dots)$. For some $x\in V$, let $x=u+v$ with $u\in U$ and $v\in U^\perp$. Then any $\hat{x}\in U$ satisfies

$$\|x-\hat{x}\|\geq\|x-u\|,$$

with equality iff $\hat{x}=u$. The unique best approximation of $x\in V$ by $\hat{x}\in U$ is the projection of x to U.

Proof: $\|\hat{x}-x\|^2=\|(\hat{x}-u)-(x-u)\|^2$ $\quad \hat{x}\in U \to \hat{x}-u$ is in U and orthogonal to $x-u=v\in U^\perp$
$=\|\hat{x}-u\|^2+\|x-u\|^2$

By (Thm 1.22) the best approx. is $\langle\hat{x}-x, y_n\rangle=0 \quad n=1,2,\dots$

1.8.2 Least Squares

Let $y_1,y_2,\dots y_n$ be cols. of A $\to$ $\hat{x}=Ah=\sum_{k=1}^n h_k y_k$ (same problem as in 1.8.1)

Recall the inner product of two col. vecs a, b: $\langle a,b\rangle = a^T\bar{b}=b^Ha$ and thus

$\langle\hat{x}-x, y_n\rangle = y_n^H\underbrace{(Ah-x)}_{\hat{x}-x}=0$ for $n=1,\dots,n$ or equiv. $\boxed{A^H(Ah-x)=0 \to A^HAh=A^Hx}$

1.8.3 LMMSE

The inner product $\langle X,Y\rangle$ of two r.v. is $E[X\bar{Y}]$ thus $\boxed{E[(\hat{X}-x)\bar{Y}_n]=0}$

$$\langle\hat{X}-X, Y_n\rangle = E\Big[\underbrace{\Big(\sum_{k=1}^n h_k Y_k - X\Big)}_{\hat{X}-X}\cdot\bar{Y}_n\Big]=0 \quad \sum_{k=1}^n E[Y_k\bar{Y}_n]h_k = E[X\bar{Y}_n] \quad n=1,\dots,n$$

in matrix notation:
$$\begin{bmatrix} E[\bar{Y}_1Y_1] & E[\bar{Y}_1Y_2] & \cdots & E[\bar{Y}_1Y_n] \\ \vdots & & & \vdots \\ E[\bar{Y}_nY_1] & E[\bar{Y}_nY_2] & \cdots & E[\bar{Y}_nY_n] \end{bmatrix}\cdot\begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix}=\begin{bmatrix} E[X\bar{Y}_1] \\ E[X\bar{Y}_2] \\ \vdots \\ E[X\bar{Y}_n] \end{bmatrix}, \quad E[\bar{Y}Y^T]h=E[X\bar{Y}]$$
$$\boxed{h^TE[YY^H]=E[XY^H]}$$

Common 'trick' for offset: $\hat{X}=h_0+\sum_{k=1}^n h_kY_k=\sum_{k=0}^n h_kY_k$ with $Y_0\triangleq 1$

Least squares / LMMSE Examples

Ex1.15 X, Z indep. real r.v. $E[Z]=0$ observe $Y=X+Z \rightarrow \hat{X}=hY$ find h LMMSE

Sol.: $E[(\underbrace{hY}_{\hat{X}}-X)Y]=0$ for $h=1$: $E[(hY-X)Y]=0 = hE[Y^2]-E[XY] \rightarrow hE[Y^2]=E[XY]$

$$\left.\begin{aligned} E[Y^2] &= E[(X+Z)^2] = \dots = E[X^2]+E[Z^2] \\ E[XY] &= E[(X+Z)X] = \dots = E[X^2] \end{aligned}\right\} \quad h=\frac{E[X^2]}{E[X^2]+E[Z^2]}$$

Prob2.8 X: r.v. with m_X, σ_X^2. Observe $Y=aX+W$ $a\in\mathbb{R}$ $m_W=0$ σ_W^2 X id. W. Compute LMMSE with offset $\rightarrow E[(X-(Yh_1+h_0)^2]$ as small as possible

Sol.: $E[(\hat{X}-X)\overline{Y_h}]=0$ write $Y_E=(1,Y)^T$ $\hat{X}=h_0+h_1Y=(h_0,h_1)Y_E=h^TY_E$

$E[Y_EY_E^T]h=E[XY_E]$

$$E[Y_EY_E^T]=E\begin{bmatrix}1 & Y\\ Y & Y^2\end{bmatrix}=\begin{pmatrix}1 & am_X\\ am_X & a^2(m_X^2+\sigma_X^2)+\sigma_W^2\end{pmatrix}$$

$$E[XY_E]=E\begin{bmatrix}X\\ XY\end{bmatrix}=\begin{pmatrix}m_X\\ a(m_X^2+\sigma_X^2)\end{pmatrix}$$

$$h_0=\frac{m_X\sigma_W^2}{a^2\sigma_X^2+\sigma_W^2}$$

$$h_1=\frac{a\sigma_X^2}{a^2\sigma_X^2+\sigma_W^2}$$

Prob2.9a) Formulate Least Squares problem: Fit a straight line through $(p_1^{(1)},p_2^{(1)}),\dots(p_1^{(m)},p_2^{(m)})$ i.e. find h_0,h_1 s.t. $\sum_{k=1}^{m}(p_2^{(k)}-f(p_1^{(k)}))^2$ with $f(x)=h_0+h_1x$ is minimal

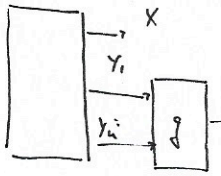
Sol.: $y^{(k)} \triangleq (1,p_1^{(k)})^T$ $h \triangleq (h_0,h_1)^T$ $f(p_1^{(k)})=h_0+h_1p_1^{(k)}=(y^{(k)})^Th$

$x \triangleq (p_2^{(1)},\dots,p_2^{(m)})^T$ $A \triangleq \begin{pmatrix}y^{(1)T}\\ y^{(2)T}\\ \vdots\\ y^{(m)T}\end{pmatrix}=\begin{pmatrix}1 & p_1^{(1)}\\ 1 & p_1^{(2)}\\ & \vdots\\ 1 & p_1^{(m)}\end{pmatrix}$ $\sum_{k=1}^{m}(p_2^{(k)}-f(p_1^{(k)})^2=\|x-Ah\|^2$ which is a LS-problem

Chapter 3: Learning Linear Functions and More About Least-Squares

3.1 Linear-Function Learning (Linear Regression)

3.1.1 Problem and Solution



Make linear estimation of X from observations $Y_1, \dots, Y_n$: $\hat{X} = \sum_{k=1}^{n} h_k Y_k$.
How to choose h? From 1.8.3 we could use $E[X\bar{Y}_c]$ and $E[Y_k \bar{Y}_c]$
but instead we are now given samples / training data:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)}) \qquad y^{(k)} \triangleq (y_1^{(k)}, \dots y_n^{(k)})^T$$

We use the vector notation $h \triangleq (h_1, \dots, h_n)^T$ $\quad y \triangleq (y_1, \dots, y_n)^T$ $\quad \boxed{g(y) = \sum_{k=1}^{n} h_k y_k = h^T y = y^T h}$

Now choose h s.t. the average squared error (ASE) is minimized

$$\boxed{ASE = \frac{1}{m}\sum_{k=1}^{m} |x^{(k)} - g(y^{(k)})|^2 = \frac{1}{m}\sum_{k=1}^{m} |x^{(k)} - (y^{(k)})^T h|^2}$$

Writing $x \triangleq (x^{(1)}, \dots, x^{(m)})^T$, $A \triangleq \begin{pmatrix} (y^{(1)})^T \\ \vdots \\ (y^{(m)})^T \end{pmatrix}$ this is a least squares problem: $\boxed{m \cdot ASE = \|x - Ah\|^2}$ (3.8)

$\hookrightarrow$ $A^H A h = A^H x$ (3.9)

3.1.2 Connection with LMMSE

Assume a statistical setting, where the samples $(x^{(k)}, y^{(k)})$ are drawn from a $p(x,y)$ independently. A and x defined as in (3.8) then $A^H x = \sum_{\ell=1}^{m} \overline{y^{(\ell)}} x^{(\ell)}$ and $A^H A = \sum_{\ell=1}^{m} \overline{y^{(\ell)}} (y^{(\ell)})^T$ are random $\left(\underbrace{A^H A}_{\text{r.v.}} h = \underbrace{A^H x}_{\text{r.v.}}\right)$. By the strong law of large numbers:

$$\lim_{m\to\infty} \frac{1}{m} A^H x = E\left[\sum \overline{y^{(\ell)}} x^{(\ell)}\right] = E[\bar{Y} X] \qquad \lim_{m\to\infty} \frac{1}{m} A^H A = E\left[\sum \overline{y^{(\ell)}} (y^{(\ell)})^T\right] = E[\bar{Y} Y^T] \quad \text{w.p.1}$$

It follows that $A^H A h = A^H x$ converges to $E[\bar{Y} Y^T] h = E[\bar{Y} X]$ which coincides with LMMSE.

> In a statistical setting, lin-func. learning converges to LMMSE estimation in the limit of infinitely many training samples.

C Gaussian Random Variables

$x \in \mathbb{R}$: $f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m_x)^2}{2\sigma^2}}$ $\qquad z \in \mathbb{C}$: $f_Z(z) = \frac{1}{\pi\sigma^2} e^{-\frac{|z-m|^2}{\sigma^2}}$ $\qquad z = x+iy$: $f_Z(x+iy) = f_X(x) f_Y(y) = \left(\frac{1}{\sqrt{2\pi\frac{\sigma^2}{2}}}\right)^2 e^{\frac{-(x-m_x)^2-(y-m_y)^2}{\sigma^2}}$

A general Gaussian random vector $X = (X_1, \dots, X_n)^T$ may be written as $\boxed{X = AU + m}$
m: mean vector $\quad A$: $\mathbb{R}/\mathbb{C}$ non-singular $n \times n$ matrix $\quad U = (U_1, \dots, U_n)^T$ indep. $\mathbb{R}/\mathbb{C}$ Gaussian r.v. with $m_U = 0$ $\sigma_U = 1$

The covariance matrix $Cov(X) \triangleq V \triangleq AA^H$, $\boxed{f_X(x) \propto f_U(A^{-1}(x-m))}$ $\qquad \boxed{f_U(u) \propto \prod_{k=1}^{n} e^{-\beta|u_k|^2} = e^{-\beta\|u\|^2}}$ $\qquad \beta = \begin{cases} 1/2 & \text{real} \\ 1 & \text{complex} \end{cases}$

It follows: $\boxed{f_X(x) \propto e^{-\beta\|A^{-1}(x-m)\|^2} = e^{-\beta(A^{-1}(x-m))^T A^{-1}(x-m)} = e^{-\beta(x-m)^H (AA^H)^{-1}(x-m)}}$ $\quad V = AA^H$, $V^{-1} = (AA^H)^{-1}$ } both pos. definite

With $W = V^{-1} = (AA^H)^{-1}$ the exponent is in a general quadratic form $q(x) = (x-m)^H W (x-m) + c$
$= x^H W x - 2\mathrm{Re}(x^H W m) + m^H W m + c$

Thm C.1 Let Y, Z be jointly Gaussian r.v. Then $X \triangleq Y + Z$ is also Gaussian.

Splitting x into two components (y,z) $q(x)$ becomes

$$q(y,z) = \left((y-m_y)^H, (z-m_z)^H\right)\begin{pmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2}\end{pmatrix}\begin{pmatrix} y-m_y \\ z-m_z\end{pmatrix} + \text{const} \quad , \quad W_{2,1} = W_{1,2}^H \qquad (C.17)$$

Thm C.2 (Gaussian Conditioning Thm) If Y,Z jointly Gaussian with joint dist $\propto e^{-\beta q(y,z)}$, then conditioned on $Z=z$ Y is Gaussian with mean

$$m_{Y|Z=z} \triangleq E[Y|Z=z] = m_Y - W_{1,1}^{-1} W_{1,2}(z-m_Z)$$

and covariance matrix $W_{1,1}^{-1}$

Thm C.3 X,Y jointly gaussian, the MAP estimate of X from $Y=y$ is an affine (= lin. with offset) function of y and coincides both with the MMSE and CMMSE (w/ offset) estimate.

Thm C.4 (Gaussian Max/Int Thm) Let $q(y,z)$ be a quadratic form as in (C.17) with $W_{1,1}$ positive definite. Then $\int_{-\infty}^{\infty} e^{-q(y,z)} dy \propto \max_y e^{-q(y,z)} = e^{-\min_y q(y,z)}$

Thm C.5 (Sum of Quadratic Forms) Let A,B pos. semi-def. matrices. Then

$$(x-a)^H A(x-a) + (x-b)^H B(x-b) = x^H W x - 2\mathrm{Re}(x^H W m) + m^H W m + c$$

$$W = A+B \qquad m = (A+B)^{\#}(Aa+Bb) \qquad c = (a-b)^H A(A+B)^{\#} B(a-b)$$

$$Wm = Aa+Bb$$

3.2 Solving Least-Squares Problems

Problem: Find $h \in F^n$ s.t. $\|Ah-x\|^2$ is minimal. $A \in F^{m\times n}$ $x \in F^m$ (3.17)

h minimizes iff $A^TAh = A^Tx$

3.2.1 Least Squares by Steepest Descent or gradient descent

Compute gradient of (3.17): $\frac{\partial}{\partial h_\ell}\|Ah-x\|^2 = \sum_{k=1}^{m}\frac{\partial}{\partial h_\ell}(A_{k,\cdot}h - x_k)^2 = \dots = 2\sum_{k=1}^{m}(A^T)_{\ell,k}(A_{k,\cdot}h - x_k) = 2(A^T)_{\ell,\cdot}(Ah-x)$

$$\hookrightarrow \nabla_h\|Ah-x\|^2 \triangleq \left(\frac{\partial}{\partial h_1}, \dots, \frac{\partial}{\partial h_m}\right)^T = 2A^T(Ah-x)$$

Algorithm: $h^{(k+1)} = h^{(k)} + \beta^{(k)}A^T(x - Ah^{(k)})$ with some initial $h^{(0)}$ e.g. $h^{(0)}=0$ (3.26)

Step size $\beta^{(k)} > 0$

Thm 3.1 For sufficiently small[1] $\beta > 0$ (3.26) with constant $\beta^{(k)} = \beta$ converges to the optimum

1): $0 < \beta < \frac{2}{\sigma^2_{max}}$ σ^2_{max} is the largest eigenvalue of A^TA.

3.2.2 Stochastic Gradient Descent

If rows of A are sufficiently random, replace (3.26) by

$$h^{(k+1)} = h^{(k)} + \beta^{(k)}(x_k - A_{k,\cdot}h^{(k)})(A_{k,\cdot})^T$$

$\rightarrow$ In each step use rough estimate of the gradient.

Generally: $h^{(k+1)} = h^{(k)} + \beta^{(k)}\sum_{\nu\in J_k}(x_\nu - A_{\nu,\cdot}h^{(k)})(A_{\nu,\cdot})^T$ $J_1 = \{1,\dots,M\}$, $J_2 = \{M+1,\dots,2M\}$ $M \ll m$

additionally:
- shuffle rows of A
- Smoothing step: $\hat{h}^{(k)} = \frac{1}{L+1}\sum_{j=k-L}^{k} h^{(j)}$ $L>0$ window parameter

3.2.3 Speedup by Momentum Term

$$h^{(k+1)} = h^{(k)} + \beta^{(k)} A^T (x - A h^{(k)}) + \alpha (h^{(k)} - h^{(k-1)})$$

Smoothes trajectory and allows larger step size $\beta^{(k)}$ and faster convergence.

3.3 LMMSE by Stochastic Gradient Descent (LMS Algorithm)

Statistical setting from 3.1.2 with samples drawn from probability law $p(x,y)$.
We wish to optimize $E[(\hat{X}-X)^2]$ with $\hat{X} = \sum_{\ell=1}^{n} h_\ell Y_\ell$

$$\frac{\partial}{\partial h_\ell} E[(\hat{X}-X)^2] = \dots = 2E[(\hat{X}-X)Y_\ell]$$

The update rule becomes

$$h^{(k+1)} = h^{(k)} + \beta^{(k)} E[(X - \hat{X}^{(k)})(Y_1, \dots, Y_n)^T]$$

3.4 Overfitting

Problem: We have too little training data for too many parameters. This results in a perfect MSE in training but fails on new data.
I.e. if A has the form $A = [\square\ \square\ 0\ \dots\ 0]$ more par. than samples

Example: $Y_1 = X + W_1$ $Y_2 = W_2$ estimate $\hat{X} = h_1 Y_1 + h_2 Y_2$ W_1, W_2 zero mean
With LMMSE we get $h_2 = 0$ $h_1 = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_{W_1}^2}$

If we don't have statistics but samples: $(x^{(1)}, y_1^{(1)}, y_2^{(1)}), (x^{(2)}, y_1^{(2)}, y_2^{(2)}) \to A = \begin{pmatrix} y_1^{(1)} & y_2^{(1)} \\ y_1^{(2)} & y_2^{(2)} \end{pmatrix}$ $x = \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix}$
inverting A and solve for h yields $h_1 = 1$ $h_2 = -\frac{\omega_1^{(1)} + \omega_1^{(2)}}{\omega_2^{(1)} + \omega_2^{(2)}}$. $|h_2| \to \infty$ for $\omega \to 0$.

Solution: L1 and L2 regularized least squares

3.5 Regularized Least Squares (L2 & L1)

3.5.1 L2-regularized Least Squares

Problem: For given matrix A and col vec x we wish to minimize $\|Ah - x\|^2$
Idea: It often turns out that overfitting is manifested in too large h_ℓ's
Solution: Modify least squares problem to minimizing $\|Ah - x\|^2 + \lambda^2 \|h\|^2$
Can be rearranged to a standard least squares problem (L2-regularized LS)

$$\|Ah - x\|^2 + \lambda^2 \|h\|^2 = \left\| \begin{pmatrix} Ah - x \\ \lambda h \end{pmatrix} \right\|^2 = \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} h - \begin{pmatrix} x \\ 0 \end{pmatrix} \right\|^2 = \|A' h - x'\|^2$$

$A' \triangleq \begin{pmatrix} A \\ \lambda I \end{pmatrix}$ $x' \triangleq \begin{pmatrix} x \\ 0 \end{pmatrix}$

3.5.2 L1-regularized LS (LASSO)

Solution: Replace LS by $\|Ah - x\|^2 + \lambda \sum_{\ell=1}^{n} |h_\ell|$ (3.55)

L1 (LASSO): "automatic feature extraction"

1. With proper λ, the h that minimizes (3.55) tends to be sparse, i.e. only few nonzero components in h
2. (3.55) is convex in h and quite easily minimized numerically.

Thm 3.2 Let $f: \mathbb{R}^n \to \mathbb{R}$, $f(h) \geq 0$ $\forall h \in \mathbb{R}^n$), differable with cont. derivatives $\frac{\partial f}{\partial h_\ell}$ at $h=0$. For $\lambda > 0$ let
shows 1. $\hat{h}(\lambda) \triangleq \operatorname{argmin}_h \left(f(h) + \lambda \sum_{\ell=1}^{n} |h_\ell| \right)$ Then for each $\ell \in \{1, 2, \dots, n\}$ $\exists$ a finite threshold $\lambda_\ell \in \mathbb{R}$ s.t. $\hat{h}_\ell(\lambda) = 0$ for $\lambda > \lambda_\ell$

Stuff:
- Nonzero h_ℓ: corresponding cols of A are support vectors
- First run LASSO to find support vectors, then run $\|Ah - x\|$ with only these in A' and x'
- Stochastic Grad. Descent: $h^{(k+1)} = h^{(k)} + \beta (x^{(k)} - \hat{x}^{(k)}) y^{(k)} - \beta \frac{\lambda}{2} \operatorname{sgn}(h^{(k)})$

Chapter 4: SVD and PCA

4.1 SVD: Singular-Value Decomposition

Notation: $z \in \mathbb{C}$: $\bar{z}$ is its complex conjugate

A matrix/vector: A^T is its transpose $\quad A^H \triangleq \overline{A^T}$ transpose and complex conjugate

A real square matrix is orthogonal if $AA^T = I$ i.e. $A^{-1} = A^T$

A complex square matrix is unitary if $AA^H = I$ i.e. $A^{-1} = A^H$

If A is (orthogonal or) unitary, then $\|Ax\| = \|x\|$ for all vectors x

$(\|Ax\|^2 = (Ax)^H(Ax) = x^H A^H A x = x^H x = \|x\|^2)$

rows ↓ ↓ # cols

$A \in \mathbb{C}^{m \times n}$
$U \in \mathbb{C}^{m \times m}$
$S \in \mathbb{R}^{m \times n}$
$V \in \mathbb{C}^{n \times n}$ (same as A)

Thm 4.1 (SVD Thm) A $\mathbb{R}$ or $\mathbb{C}$ matrix A can be written as $\boxed{A = USV^H}$ where

- U and V are unitary (real and orthogonal if A is real)
  - unitary is complex and $UU^H = I$, U, V are square
- S is diagonal with nonnegative real elements $S = \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{bmatrix}$ σ_i: singular values
- condition number $= \frac{\sigma_{max}}{\sigma_{min}}$ of a nonsingular square matrix $\quad V^H V = I \quad U^H U = I$

Example: $A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$ SVD: $A = USV^H = \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{bmatrix}}_{V^H}$

Thm 4.2 (Singular values and eigenvalues) Let $A \in \mathbb{R}^{m \times n} / \mathbb{C}^{m \times n}$ $A = USV^H$ u_ℓ: ℓ-th col of U, v_ℓ: ℓ-th col of V

$\boxed{AA^H u_\ell = \sigma_\ell^2 u_\ell}$ $\quad$ $\boxed{A^H A v_\ell = \sigma_\ell^2 v_\ell}$

cols of U are eigvects of AA^H
cols of V are eigvects of $A^H A$
$\boxed{\text{eigenvalues } \lambda_\ell = \sigma_\ell^2}$

4.2 Pseudo-Inverse

The Moore-Penrose pseudo-inverse of $A \in \mathbb{R}/\mathbb{C}$ (not necessarily square) with SVD $A = USV^H$ and non-zero singular values σ_ℓ is

$$A^\# \triangleq V S^\# U^H \qquad S^\# \triangleq \begin{bmatrix} \text{diag}(\sigma_1^{-1}, \dots, \sigma_\ell^{-1}) & 0 \\ 0 & 0 \end{bmatrix}$$

$S^\#, A^\#$ same dim. as S^H, A^H; zero σ_ℓ are not inverted

Ex 4.2 $A = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{U} \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{S} \underbrace{(1)}_{V^H}$ $\qquad A^\# = \underbrace{(1)}_{V} \underbrace{(1\ 0)}_{S^\#} \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{U^H} = (1\ 0)$

Usage: - Compute Least Squares "best fit" to a system of lin. eq. that lacks a unique solution

Thm 4.3 $A \in \mathbb{R}^{m \times n} / \mathbb{C}^{m \times n}$

$$\begin{aligned} &AA^\# A = A \qquad && (AA^H)^\# = (A^\#)^H A^\# \\ &A^\# A A^\# = A^\# \qquad && A^\# x = 0 \iff x^H A = 0 \iff A^H x = 0 \\ &(A^H)^\# = (A^\#)^H \end{aligned}$$

$(ABC)^H = C^H B^H A^H$

Thm 4.4 (Pseudo-Inverse in Closed Form)

If $\text{rank}(A) = \#\text{rows}$	$\longrightarrow$ $AA^\# = I$	$A^\# = A^H (AA^H)^{-1}$
If $\text{rank}(A) = \#\text{cols}$	$\longrightarrow$ $A^\# A = I$	$A^\# = (A^H A)^{-1} A^H$

In general however, there is no closed-form sol. for the pseudo-inverse.

4.3 Projection to the Column Space

If the cols of $A \in \mathbb{R}^{m\times n}/\mathbb{C}^{m\times n}$ are orthogonal, then the projection to the space spanned by them is $\mathbb{F}^m \to \mathbb{F}^m : x \mapsto AA^H x$ (in space → out space; function / mapping; input arguments)

If not orthogonal, we have Thm 4.5

Thm 4.5 (Projection by SVD) For any $m\times n$ matrix A over $\mathbb{F}=\mathbb{R}/\mathbb{C}$ the linear mapping

(4.44) $\boxed{\mathbb{F}^m \to \mathbb{F}^m : x \mapsto AA^H x}$ $\quad A = USV^H \quad A^\# \triangleq VS^\# U^H$

$\#$ is the projection of x to the subspace spanned by the (not necessarily orthogonal) cols of A. If $A=USV^H$ (4.44) can also be written as $\boxed{\mathbb{F}^m \to \mathbb{F}^m : x \mapsto U_t U^H x}$ where U_t consists of the cols of U that correspond to ~~the~~ nonzero singular values.

Thm 4.6 (SVD Projection Energy) Let $y_1 \dots y_n$ cols of A over $\mathbb{F}=\mathbb{R}/\mathbb{C}$ $A\in\mathbb{F}^{m\times n}$, $A=USV^H$, σ_i sing. values

For all cols u_k of U: $\boxed{\sum_{l=1}^{n} |\langle y_l, u_k\rangle|^2 = \sum_{l=1}^{n} |u_k^H y_l|^2 = \sigma_k^2}$

Each squared sing. val. σ_k^2 is the total energy of the projections of $y_1,\dots,y_n$ to the 1D space spanned by u_k

4.4 SVD and Least Squares

4.4.1 Pseudo-Inverse sol. of LS

Recall: LS-problem: $\hat{h} = \arg\min_h \|Ah-x\|^2$ (4.52) iff $A^HAh = A^Hx$ (4.53) $\to h = (A^HA)^{-1}A^Hx$

Thm 4.7 (Pseudo-Inv. sol. of LS) The vector $\boxed{h = A^\# x}$ is a sol. of (4.53) ie minimizes $\|Ah-x\|^2$.

Moreover, if h' also minimizes $\|Ah-x\|^2$, then $\|h\| \le \|h'\|$ with equality only if (iff) $h'=h$

4.4.2 Singular Values in Least Squares

From Thm 4.7 with $A = USV^H = U\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & \vdots\\ 0 & \cdots & \end{bmatrix}$ is $h = A^\# x = V\underbrace{\begin{bmatrix}\sigma_1^{-1} & 0 & 0\\ 0 & \sigma_2^{-1} & \vdots\\ 0 & \cdots & \end{bmatrix}}U^H x$

↳ Small singular values σ_i have large impact on h. This typically happens with random square matrices.

Overfitting is manifested in such large values of h. One way to alleviate this is setting $\sigma_e^{-1} = 0$ for too small σ_e.

4.5 Principle-Components Analysis (PCA)

$\mathbb{F}=\mathbb{R}/\mathbb{C}$

Given: Some data in col. vectors $y_1,\dots,y_N \in \mathbb{F}^M$, mostly stripped off its mean s.t. $\sum_{k=1}^{N} y_k = 0$

Problem: For some K, $\boxed{1\le K<M}$ wish to find K-dimensional subspace of $\mathbb{F}^M$ that contains as much of the "energy" of data as possible. $\hat{y}_n$ is proj. of y_n, $\boxed{\sum_{n=1}^{N}\|\hat{y}_n\|^2}$ as large as possible or equiv. $\sum_{n=1}^{N}\|y_n\|^2 - \sum_{n=1}^{N}\|\hat{y}_n\|^2$ as small as possible.

Ansatz: Let B be a mtx over $\mathbb{F}$ whose cols form an orthonormal basis of the desired subspace. $B\in\mathbb{F}^{M\times K}$ and $\boxed{B^HB = I_K}$ (orthonormality). The projection to span(cols of B): $\boxed{\mathbb{F}^M \to \mathbb{F}^M : y \mapsto \hat{y} = BB^H y}$

Total energy of projection: $\boxed{\sum_{n=1}^{N}\|\hat{y}_n\|^2 = \sum_{n=1}^{N}\|BB^H y_n\|^2 = \sum_{n=1}^{N}\|B^H y_n\|^2}$ (4.74), (4.75)

Ansatz (cont.): Now let $A \triangleq (y_1, \dots, y_N) \in \mathbb{F}^{M \times N}$ matrix with "data" and $A = USV^H$ its SVD.
Assume $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$

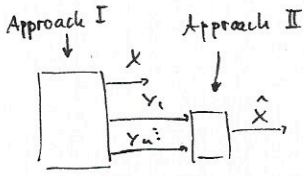
Thm 4.9 (SVD-PCA Theorem) The $M \times K$ mtx $\boxed{B \triangleq U \begin{pmatrix} I_K \\ 0 \end{pmatrix}}$ (the first K cols of U) maximizes (4.74) among all $M \times K$ matrices with orthonormal columns.

Chapter 5: Some Basics

5.1 Parametrized Estimators vs. Global statistical Models

What is given	What we can apply
Full statistical model $\boxed{p(x,y)}$	MAP ML MMSE
Partial model $\boxed{p(y\|x)}$	ML some penalized likelihood

Goal: Approximate x from $y_1, \dots y_M$



$p(x|y) = \frac{p(x,y)}{p(y)} \propto p(y|x)p(x)$

If no statistical information is given, we need $\boxed{\text{samples } (x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})}$

Naive approach $\mathcal{X}, \mathcal{Y}$ sets of possible values of x, y. If $\mathcal{X}, \mathcal{Y}$ finite with not too many elements, approximate $p(x,y)$ by $p_S(x,y)$ the empirical distribution of training samples

$p_S(x,y) \triangleq \frac{1}{m} \times$ (#occurrences of (x,y) in training samples) $\lim_{m \to \infty} p_S(x,y) = p(x,y)$ w.p.1 (law of large numbers)

Requires that each pair appears sufficiently often.

Approach I: Parametrized Global Statistical Models: = "generative model"

Assume the unknown $p(x,y)$ belongs to a family of prob. laws $p(x,y|\theta)$ (parameters). Based on training data, form an estimate $\hat{\theta}$ of θ. Then use $p(x,y|\hat{\theta})$ to estimate x.

$$\hat{\theta} = \operatorname*{argmax}_{\theta} p(x^{(1)}, y^{(1)}, \dots, x^{(m)}, y^{(m)} | \theta)$$

Treated in Part III

Approach II: Parametrized Estimators

The estimation function $g: y \mapsto \hat{x}$ belongs to a family of functions $\{g_\theta\}$ with parameter θ. Based on training data, estimate $\hat{\theta}$ which determines the estimator.

Example: $g: (y_1, \dots y_M) \mapsto \sum_{k=1}^{M} h_k y_k$

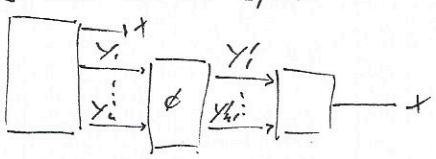
Neural networks → Chapter 6 Kernel methods → Chapter 7

Nonlinear estimators → following

5.2 Features, Transforms, and Lifted Linear Estimation

Before estimating, transform observations $Y = (Y_1 \dots Y_M)^T$ into $\phi(Y) = Y' = (Y'_1 \dots Y'_{M'})^T$. Dimensions need not be equal

Benefits:
- Extract features that simplify estimation
- Reduce dimensionality to a manageable size
- Transform to higher dimension in which esti. is easier
- Transform nonnumerical Y to numerical Y'



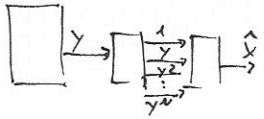
$Y' \mapsto \hat{X} = \sum_{k=1}^{M'} h_k Y'_k$ $\quad x \triangleq (x^{(1)}, \dots, x^{(m)})^T$ $\quad h \triangleq (h_1, \dots, h_{M'})^T$ $\quad A \triangleq \begin{pmatrix} \phi(y^{(1)})^H \\ \vdots \\ \phi(y^{(m)})^H \end{pmatrix} = \begin{pmatrix} y'^{(1)}_1 & y'^{(1)}_2 & \dots & y'^{(1)}_{M'} \\ \vdots & & & \vdots \\ y'^{(m)}_1 & \dots & & y'^{(m)}_{M'} \end{pmatrix}$

$ASE \triangleq \frac{1}{m} \sum_{k=1}^{m} |x^{(k)} - \phi(y^{(m)})h|^2 = \frac{1}{m} \|x - Ah\|^2$

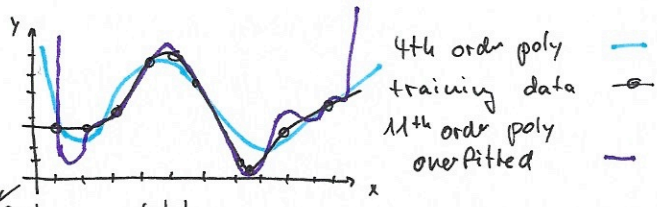
$\boxed{\text{Is a Least Squares Problem}}$

Example 5.1 Polynomial fitting

$\phi: \mathbb{R} \to \mathbb{R}^{M+1}: y \mapsto (1, y, y^2, \dots, y^M)^T$ $\quad \hat{X} = \phi(Y)^T h$



5.3 On Overfitting and Regularization



Problem: Complex models can be adjusted to may fit the training data very well but fail completely on new data.

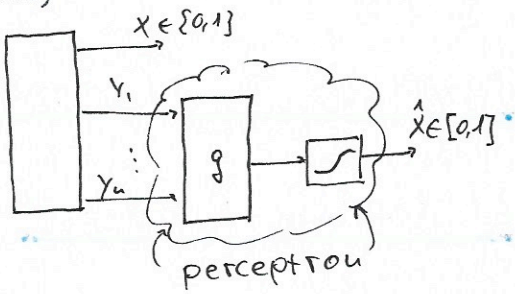
General Solution:
1. Split data into two parts
2. Train models on different complexity on first half
3. Compare performance of trained models on second half of data. Select the best performing model

Leave-one-out:
1. Train model on all data ~~expt~~ except 1 sample
2. Do this for each sample as the excluded one
3. Test error on excluded sample
4. Final score is the average error over all repetitions

5.4 Learning Classifiers and Class Probabilities (Logistic Regression)

Assumption: X is a $\{0,1\}$-valued variable. Eg. an indicator of some event or class or condition of interest

Idea: Rather than producing a $\{0,1\}$ estimate of X, produce a $[0,1]$-valued $\hat{X}$ that may be interpreted as estimate/guess of



Soft estimate/ soft classifier → $P(X=1 \mid Y=y) = E[X \mid Y=y]$ (5.17)

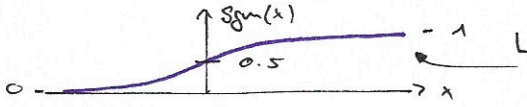
only bcs $X \in \{0,1\}$

5.4.1 Turning Generic Parametrized Estimators into Soft Classifiers

Given: A generic parametrized mapping $g: \mathbb{R}^n \to \mathbb{R}$ **Required:** Soft classifier s.t. $\hat{X} \in [0,1]$

Idea: Append g with a function that limits to $[0,1]$ with monotonous increase

E.g. $\operatorname{sgm}(x) \triangleq \frac{1}{1+e^{-x}}$



Logistic sigmoid function
mapping of this form is called a perceptron

Example: g might be an affine function → Soft estimate is $\operatorname{sgm}(h_0 + h_1 Y_1 + h_2 Y_2 + \cdots + h_n Y_n)$

Note For any prob. dist $p(x,y)$ with $\{0,1\}$-valued X $\quad P(X=1 \mid Y=y) = \frac{1}{1+e^{-L_X(y)}}$

with $L_X(y) \triangleq \ln \frac{P(X=1 \mid Y=y)}{P(X=0 \mid Y=y)} = \ln \frac{P(y \mid X=1)}{P(y \mid X=0)} + \ln \frac{P(X=1)}{P(X=0)}$

5.4.2 Learning by minimizing the MSE

Learning by MSE minimization works also in this case:

$$E[MSE] = E\left[\frac{1}{m}\sum_{i=1}^{m} |x^{(i)} - \hat{x}(y^{(i)})|^2\right] = \underbrace{E[|X - \hat{x}(Y)|^2]}_{(5.25)} \qquad \lim_{m\to\infty} MSE \overset{\text{wp1}}{=} E[MSE]$$

(5.25) is minimized by $\hat{x}(y) = E[X \mid Y=y]$ which is (5.17)

- Probabilities close to one or zero are learned with poor precision
- Minimizing MSE is <u>not a (linear) least-squares problem</u>, even if g is linear

5.4.3 Discriminative Statistical Modeling and ML Learning

Approach: Use statistical models, where samples $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$ are drawn indep. from a dist. $p(x,y|\theta)$ s.t. $p(x,y|\theta) = p(x|y,\theta)\,p(y)$ ($p(y)$ assumed not dep. on θ)

Further we define $p(x|y,\theta) \triangleq \begin{cases} \mathrm{sgm}(g(y)) & x=1 \\ 1-\mathrm{sgm}(g(y)) & x=0 \end{cases}$ g a fct. with parameters θ

ML estimation: Now we fit this statistical model to the given samples

$$\hat\theta = \operatorname*{argmax}_\theta \prod_{\ell=1}^{m} p(x^{(\ell)}, y^{(\ell)}|\theta) = \dots = \operatorname*{argmax}_\theta \frac{1}{m}\sum_{\ell=1}^{m}\underbrace{\left(x^{(\ell)}\ln\left(\mathrm{sgm}(g(y^{(\ell)}))\right) + (1-x^{(\ell)})\ln\left(1-\mathrm{sgm}(g(y^{(\ell)}))\right)\right)}_{-K(\theta)\ \leftarrow\ \text{cost fnction}} \qquad (5.33)$$

Minimizing $K(\theta)$ can be done by (stochastic) gradient descent or other methods.

Special case: $g(y) = h_0 + h_1 y_1 + \dots + h_n y_n$ $\quad \theta = h \triangleq (h_0, \dots, h_n)^T$ The gradient ∇_θ of K with respect to $\theta = h$

$$\nabla_\theta (K(h)) = \frac{-1}{m}\sum_{\ell=1}^{m}\left(x^{(\ell)}(1-\mathrm{sgm}(h^T y^{(\ell)})) + (1-x^{(\ell)})\,\mathrm{sgm}(h^T y^{(\ell)})\right)$$

Chapter 6: Neural Networks

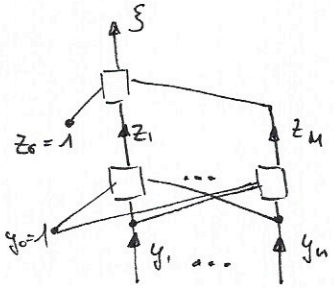
Most successful type: Variations of multilayer perceptrons.

6.1 Multilayer Perceptrons

6.1.1 Structure

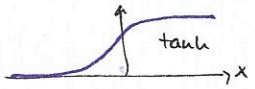
Two-layer perceptron: $\mathbb{R}^n \to \mathbb{R} : (y_1, \dots, y_n) \mapsto \xi(y_1, \dots, y_n)$

$$\xi(y_1, \dots, y_n) = g_{out}\left(w_0 + \sum_{j=1}^{M} w_j\,\tilde g\left(w_{j,0} + \sum_{i=1}^{n} w_{j,i}\,y_i\right)\right)$$

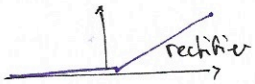


w_j, $w_{j,i}$: real parameters / "weights"

$y_0 \triangleq 1$ $\quad z_j \triangleq \tilde g\left(\sum_{i=0}^{n} w_{j,i}\,y_i\right)$ $\quad z_0 \triangleq 1$ $\quad \to \xi = g_{out}\left(\sum_{j=0}^{M} w_j z_j\right)$



$\tilde g(x) = \mathrm{sgm}(x)$ or $\tilde g(x) = \tanh(x) \triangleq \frac{e^x - e^{-x}}{e^x + e^{-x}}$ or $\tilde g(x) = \mathrm{rectifier}(x)$



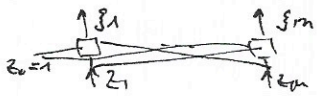
$g_{out}(x) = x$ or $g_{out}(x) = \mathrm{sgm}(x)$ if used for classification

Important: All nonlinearities are continuous and (almost[1]) everywhere differentiable, which matters when it comes to optimizing the parameters w_j and $w_{j,i}$

[1] rectifier is not at zero

Hidden nodes: $z_1, \dots, z_M$ are called hidden nodes/variables

Several outputs:



Deep networks: With many layers. Most notable class is the convolutional neural networks (CNN)

6.1.2 Learning (Training) and Regularization

m number of samples

Training means the calculation of weights in all layers from samples $(x^{(1)}, y_1^{(1)}, y_2^{(1)}, \dots, y_n^{(1)})$ $(x^{(2)}, y_1^{(2)}, y_2^{(2)}, \dots, y_n^{(2)})$ ...

Approach: (Approx.) minimize a cost function such as the ASE $= \frac{1}{m}\sum_{\ell=1}^{m} |x^{(\ell)} - \xi(y_1^{(\ell)}, \dots, y_n^{(\ell)})|^2$ on the training samples.

For class probabilities, $K(h)$ from (5.33) can be used as metric

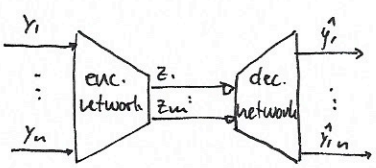
If $g_{out}(x) = x$ then optimizing w_j is a least squares problem. For the lower layers however, it is **not**.

Computing layer weights: By back-propagation (not treated). Take the derivative of the cost function with respect to the individual weights and apply a gradient descend. Can be computationally demanding

Avoiding Overfitting:
- Switch off random weights for short amount of time
- Augment training data by distortion/noise
- Early stopping if error on test data stops decreasing

6.3 Autoencoders, GANs, and Neural-Network Priors

6.3.1 Autoencoder Networks

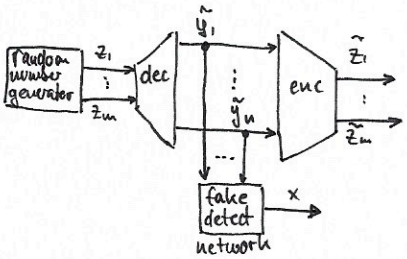


- Is trained to reproduce its input $y \in \mathbb{R}^n$ at the output $\hat{y} \in \mathbb{R}^n$
- $z = (z_1, \dots, z_m)^T \in \mathbb{R}^m$, $m \ll n$ are the hidden units
- enc/dec are some sort of multilayer perceptrons (enc. mostly CNN)
- Trained by unsupervised learning (= on unlabeled data)

Usages:
- Dim. reduction from $\mathbb{R}^n$ to $\mathbb{R}^m$, only enc. is used after training
- Compression: z is used for transmission/storage
- Denoising: Noise channel on input, noisefree at output → use enc. after training
- Learning a generative model → use dec after learning

Often m is larger than n initially, then use sparsifying cost fun (ASE L1 for example) such that z becomes sparse with $m < n$.

6.3.2 Generative Adversial Networks (GANs)



- Produces new samples $\tilde{y}^{(1)}, \tilde{y}^{(2)}, \dots$ according to complex learned statistics
- A way of training autoencoders that can improve their performance
- dec/enc are first trained as an autoencoder
- After this:
  - dec is fed with iid samples $z \in \mathbb{R}^m$ and produces $\tilde{y} \in \mathbb{R}^n$
  - Fake detect is trained to produce $x=0$ for given samples from enc/dec training y and $x=1$ for output of dec $\tilde{y}$
  - dec trained to produce $x=0$ and $\hat{z}=z$
  - enc trained to produce $\hat{z}=z$

The fake detect network tries to discriminate btw samples of given set y and produced $\tilde{y}$ by the dec

The dec tries to fool the fake detect with $\tilde{y}$ that cannot be distinguished from y. This arms race results in each of them becoming very good. The enc prevents the dec from producing $\tilde{y}=y$.

6.3.3 Neural Network Priors

$$\hat{x} = \underset{x \in \mathbb{R}^n}{\operatorname{argmax}}\, p(y|x)$$

$p(y|x)$: likelihood function

- The dec. of GANs and autoencoders can be used as priors or regularizers of statistical estimation problems.
- Constraint: x can be produced by decoder network: $f: \mathbb{R}^m \to \mathbb{R}^n$ $m \ll n$

$$\hat{x} = f(\hat{z}) \quad \hat{z} = \underset{z}{\operatorname{argmax}}\, p(y|f(z))$$

6.4 Radial-Basis Function Networks

Is a mapping $\mathbb{R}^n \to \mathbb{R}: y \mapsto \sum_{j=1}^{M} w_j \phi_j(y) + w_0$ $\quad w_j \in \mathbb{R}$ parameters/weights, $\phi_j: \mathbb{R}^n \to \mathbb{R}$ nonlinear function

$\phi_j(y)$ = some function of $\|y - \mu_j\|$, $\mu_j \in \mathbb{R}^n$ parameter/center.

Gaussian with mean vect μ_j and cov. mtx. W_j^{-1}

$$\phi_j(y) = \exp\left(-\frac{\|y-\mu_j\|^2}{2\sigma^2}\right)$$

$$\phi_j(y) = \exp\left(-\frac{1}{2}(y-\mu_j)^T W_j (y-\mu_j)\right)$$

- Training is a nonlin. opt. problem that can be approximated but no guarantee of finding global optimum
- Training is quite different from that of multilayer perceptrons. ϕ_j are trained by samples of y alone (i.e. unlabeled data), optimizing w_j is then a standard least squares problem

K-means clustering algorithm with training samples $y^{(e)}$:

1. Select $k=M$ mean vects μ_j, $j=1,\dots,M$ at random from training set $\{y^{(e)}\}$
2. Cluster the samples into classes C_j by assigning each to the nearest mean vector
$$C_j := \{y^{(e)} : \|y^{(e)} - \mu_j\| \le \|y^{(e)} - \mu_k\|, \ k \neq j\}$$
3. New mean vectors are mean of their clusters: $\mu_j := \frac{1}{|C_j|}\sum_{y \in C_j} y$
4. Repeat 2 & 3 until no change in sets C_j

works because $\sum_{j=1}^{M}\sum_{y \in C_j} \|y-\mu_j\|^2$ decreases in every iteration → also must terminate

Chapter 7: Kernel Methods

can be used for regression, classification, clustering and more

7.1 Form of Mapping

A kernel based regression function $S \to \mathbb{R}: y \mapsto \hat{x}$ has the form

$$\hat{x} = \sum_{j=1}^{m} w_j k(y, y^{(j)}) + w_0 \quad (7.1)$$

($w_j \in \mathbb{R}$ weights; $y^{(j)}$ training samples $y^{(1)},\dots,y^{(m)}$)

$k: S \times S \to \mathbb{R}$ is called the kernel. The domain S need not be in $\mathbb{R}^n$ (eg. images, strings, ...)

Example $k(y,y') = \exp\left(-\frac{\|y-y'\|^2}{2\sigma^2}\right)$ Difference with radial-basis function:
- k has no parameters to be adjusted
- Sum (7.1) runs over all training samples

Choosing a suitable k is the most important step in practical applications

7.2 Learning and Regularization

Training data $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$

$$ASE = \frac{1}{m}\sum_{l=1}^{m}\left(x^{(l)} - \sum_{j=1}^{m} w_j k(y^{(l)}, y^{(j)}) - w_0\right)^2$$ → LS-problem

coeff w_j = # training samples ⇒ can have ASE=0 by $x^{(l)} = \sum_{j=1}^{m} w_j k(y^{(l)}, y^{(j)}) + w_0$ $l=1,\dots,m$ for weights w_j, $j=0,\dots,m$

BUT: results in large weights & bad performance on new data, → Regularization needed

L1 regularization

$$ASE_{L1} = \frac{1}{m}\sum_{l=1}^{m}\left(x^{(l)} - \sum_{j=1}^{m} w_j k(y^{(l)}, y^{(j)}) - w_0\right)^2 + \lambda \sum_{j=1}^{m} |w_j| \quad (7.5)$$ ($\lambda > 0$)

can result in zero weights → complexity reduction in (7.2) which is essential in kernel methods.

Nonzero w_j → $y^{(j)}$ are called support vectors.

As of section 3.5.2., (7.5) is convex in w_j

7.3 Feature Space Interpretation and Kernel Trick

A nonlinear mapping combined with linear estimation $\hat{x} = \phi(y)^T v$ (7.6) from Section 5.2 for coeff $v \in \mathbb{R}^n$.

If $k(y,z) \triangleq \phi(y)^T \phi(z)$ (7.7) (7.6), (7.1) are essentially equivalent. (7.7) is a Mercer kernel.

Thm 7.1 (Kernel Trick Thm): $\phi: S \to \mathbb{R}^n$, $y^{(1)},\dots,y^{(m)} \in S$ given and fixed, $k(y,z) = \phi(y)^T\phi(z)$. Then for any given $w_1,\dots,w_m \in \mathbb{R}$ there exists $v \in \mathbb{R}^n$ s.t.

$$\sum_{j=1}^{m} w_j k(y, y^{(j)}) = \phi(y)^T v \quad (7.8)$$ for all $y \in S$.

Conversely, for any $v \in \mathbb{R}^n$, $\exists v' \in \mathbb{R}^n$ and $w_j \in \mathbb{R}$ s.t. $\sum_{j=1}^{m} w_j k(y, y^{(j)}) = \phi(y)^T v'$ and $\phi(y)^T v' = \phi(y)^T v$ if $\phi(y)$ is in the subspace of $\mathbb{R}^n$ spanned by $\phi(y^{(1)}),\dots,\phi(y^{(m)})$.

Summary
- Mappings of form $\hat{x} = \sum w_j k(y, y^{(j)}) + w_0$ are mappings of form $\phi(y)^T v$ in disguise ("rotkäppchen")
- Computation and their complexity are quite different
- Sparsifying by L1 doesn't yield the same mappings

Part III: Structured Statistical Models & State Space Models

Chapter 8: Factor Graphs and Hidden Markov Models

- Statistical models with many variables have usually some modular structure
- These can be represented by factor graphs
- They allow a unified approach to a wide variety of structured models and algorithms

An example modular structure are hidden Markov models.

8.1 Markov Chains and Hidden Markov Models

Def: Three r.v. X, Y, Z form a Markov Chain if, conditioned on event $Y = y$ X and Z are independent. Z depends only on Y.

i.e.: if $p(x,z \mid y) = p(x \mid y)\, p(z \mid y) \Rightarrow p(x,y,z) = p(x)\, p(y \mid x)\, p(z \mid y)$ (Generally: $p(x,y,z) = p(x)\, p(y \mid x)\, p(z \mid x,y)$)

$X \quad Y \quad Z$

- Holds for prob. mass and prob. density functions

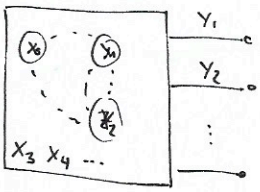
In general: $X_1, X_2, \dots, X_n$ form a MC if, for $1 < k < n$ conditioned on $X_k = x_k$ $(X_1, \dots, X_{k-1})$ and $(X_{k+1}, \dots, X_n)$ are independent. I.e. if

$$p(x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n \mid x_k) = p(x_1, \dots, x_{k-1} \mid x_k) \cdot p(x_{k+1}, \dots, x_n \mid x_k)$$

$$\Rightarrow p(x_1, \dots, x_n) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1})$$

Def (Hidden Markov Model): $X_0, X_1, X_2, \dots$ hidden variables / state variables — not directly observable

$Y_1, Y_2, Y_3, \dots$ observable variables



for all $n \geq 1$ $X_0, X_1, \dots, X_n$ form a Markov Chain and

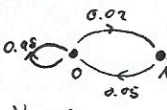
$$p(y_1, \dots, y_n \mid x_0, \dots, x_n) = \prod_{k=1}^{n} p(y_k \mid x_k, x_{k-1}) \quad (8.13)$$

It follows

$$p(x_0, \dots, x_n, y_1, \dots, y_n) = p(x_0) \prod_{k=1}^{n} p(x_k, y_k \mid x_{k-1})$$

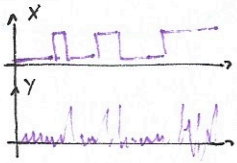
It is usually assumed that the r.v. X_i take values in some finite set

Ex 8.1 $X_0, X_1, \dots$ is a MC with $X_k \in \{0,1\}$ with transition prob. $p(x_k \mid x_{k-1})$ as in



$Y_1, Y_2, \dots$ output $\in \mathbb{R}$ behaves as in (8.13)

Y_k: Gaussian with $\mu = 0$, $\sigma^2 = 0.1^2$ if $X_{k-1} = 0$; Gaussian with $\mu = 0.25$, $\sigma^2 = 0.5^2$ if $X_{k-1} = 1$ →



Common computational problems For fixed observations $Y_1 = y_1, \dots, Y_n = y_n$

Current state estimation: $p(x_n \mid y_1, \dots, y_n) = \dfrac{p(x_n, y_1, \dots, y_n)}{p(y_1, \dots, y_n)}$ & $p(x_n, y_1, \dots, y_n) = \sum_{x_0} \cdots \sum_{x_{n-1}} p(x_0, \dots, x_n, y_1, \dots, y_n)$ (8.17)

Prediction of next output: $p(y_{n+1} \mid y_1, \dots, y_n) = \dfrac{p(y_1, \dots, y_{n+1})}{p(y_1, \dots, y_n)}$ & $p(y_1, \dots, y_{n+1}) = \sum_{x_0, \dots, x_{n+1}} p(x_0, \dots, x_{n+1}, y_1, \dots, y_{n+1})$ (8.20)

Estimation of state at any time $k \leq n$: $p(x_k \mid y_1, \dots, y_n) = \dfrac{p(x_k, y_1, \dots, y_n)}{p(y_1, \dots, y_n)}$ & $p(x_k, y_1, \dots, y_n) = \sum_{x_0, \dots, x_n \text{ except } x_k} p(x_0, \dots, x_n, y_1, \dots, y_n)$ (8.22)

MAP estimation of state trajectory: $(\hat{x}_0, \dots, \hat{x}_n)_{MAP} = \operatorname{argmax}_{x_0, \dots, x_n} p(x_0, \dots, x_n \mid y_1, \dots, y_n) = \operatorname{argmax}_{x_0, \dots, x_n} p(x_0, \dots, x_n, y_1, \dots, y_n)$ [1]

Probability of the observation: $p(y_1, \dots, y_n) = \sum_{x_0, \dots, x_n} p(x_0, \dots, x_n, y_1, \dots, y_n)$ (8.25)

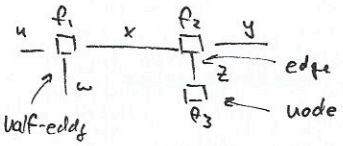
- Note their similar forms
- Brute force is infeasible

1) If $(\hat{x}_0, \dots, \hat{x}_n)$ is unique, then $\hat{x}_k = \operatorname{argmax}_{x_k} \underbrace{p_k(x_k, y_1, \dots, y_n)}_{\triangleq \max_{x_1, \dots, x_n \text{ except } x_k} p(x_0, \dots, x_n, y_1, \dots, y_n)}$

8.2 Factor Graphs

Here: Forney factor graphs, Represents the factorization of a function of several vars.

Ex $f(u,w,x,y,z) = f_1(u,w,x)\, f_2(x,y,z)\, f_3(z)$
(f: global function; f_1, f_2, f_3: factors)



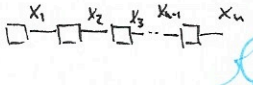
- there is a (unique) node for every factor
- there is a (unique) edge or half-edge for every variable
- node g is connected to edge x iff g is a function of x

- A configuration is a particular assignment of values to all variables.
- The configuration space is the set of all configurations, i.e. the domain of the global function.
- A configuration ω is called valid if $f(\omega) \neq 0$.
- In each configuration, the variables in the graph have a definite value. → We can consider the variables as functions of the configuration space: $X : (u,w,x,y,z) \mapsto x$

Statistical models as factor graphs

Markov Chain

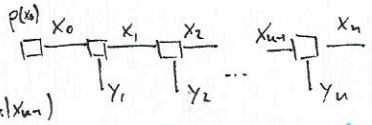
$p(x_1,\dots,x_n) = p(x_1)p(x_2|x_1)\cdots p(x_n|x_{n-1})$



(8.6)

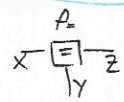
~~Hidden Markov Model~~

$p(x_0,\dots,x_n,y_1,\dots,y_n) = p(x_0)\prod_{k=1}^{n} p(y_k,x_k|x_{k-1})$

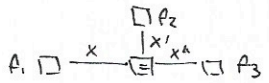


Equality check function

$f_=(x,y,z) \triangleq \begin{cases} 1, & \text{if } x=y=z \\ 0, & \text{else} \end{cases} = \delta[x-y]\,\delta[y-z]$



Enables this: $p(x) = f_1(x) f_2(x) f_3(x)$

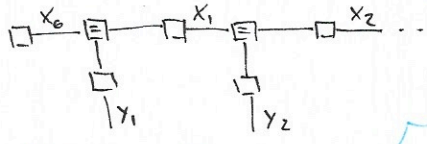


because $x = x' = x''$ holds for all valid configurations.

(8.9)

Hidden Markov Model

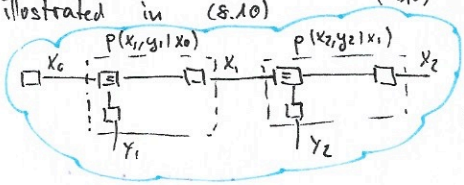
$p(x_0,\dots,x_n,y_1,\dots,y_n) = p(x_0)\prod_{k=1}^{n} p(x_k|x_{k-1})p(y_k|x_{k-1})$



Boxes

(8.9) may be viewed as a refined version of (8.6) as illustrated in (8.10)

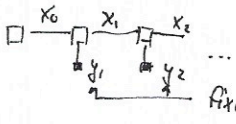
(8.10)



Constraints $f_=(x,y,z) = \delta[x-y]\,\delta[y-z]$ is an example of a constraint. A deterministic relation such as $z = g(x,y)$ may be expressed by the factor $\delta[z-g(x,y)]$

Known & Constant Variables

Example



May be plugged into the factor in which case the corresponding edges can be removed.

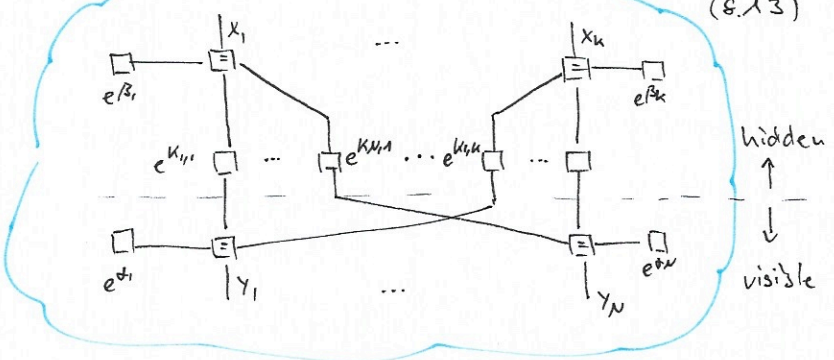
Ex 8.2 (Restricted Boltzmann Machine)

(8.13)

$$p(\underbrace{x_1,\dots,x_K}_{\text{hidden}\ \in\{0,1\}},\ \underbrace{y_1,\dots,y_N}_{\text{visible}\ \in\{0,1\}}) \propto \exp\left(\sum_{i=1}^{N}\alpha_i(y_i) + \sum_{j=1}^{K}\beta_j(x_j) + \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}(y_i,x_j)\right)$$

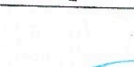
$$= \left(\prod_{i=1}^{N} e^{\alpha_i(y_i)}\right)\left(\prod_{j=1}^{K} e^{\beta_j(x_j)}\right)\left(\prod_{i=1}^{N}\prod_{j=1}^{K} e^{w_{ij}(y_i,x_j)}\right)$$

$$\alpha_i(y_i) = \begin{cases} a_i, & y_i = 1, \\ 0, & y_i = 0. \end{cases} \qquad \beta_j(x_j) = \begin{cases} b_j, & x_j = 1, \\ 0, & x_j = 0. \end{cases} \qquad w_{ij}(y_i,x_j) = \begin{cases} w_{ij}, & y_i = x_j = 1, \\ 0, & \text{else}. \end{cases}$$



8.3 On Factor Graphs of Statistical Models

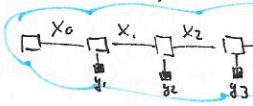
Thm 8.1 (Independent Components) If the factor graph of a joint prob. dist. $p(x,y)$ consists of two unconnected components, then every r.v. in one component is indep. of every r.v. in the other component.



Thm 8.2 (Posterior Probability) For fixed $y_1,\dots,y_N$ s.t. $p(y_1,\dots,y_N) \neq 0$ a FG of $p(x_1,\dots,x_K,y_1,\dots,y_N)$ is also a FG of $p(x_1,\dots,x_K|y_1,\dots,y_N)$ up to a scale factor. Proof: $p(A|B) = \frac{p(A,B)}{p(B)}$ & $p(A,B)$

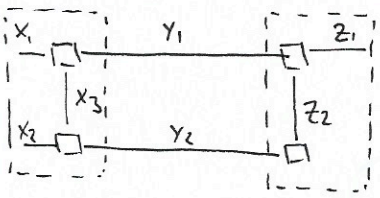
(8.12)

Ex 8.5



represents $p(x_0,\dots,x_n,y_1,\dots,y_n)$ and $p(x_0,\dots,x_n|y_1,\dots,y_n)$

Thm 8.3 (Markov Cut Theorem) FG of joint prob. function of r.v. $x_1,..,x_c, y_1,..,y_m$ and $z_1,..,z_n$. If edges of $y_1,..,y_m$ are removed and two graphs with only x in one and z in the other. Then for any fixed y s.t. $p(y_1,..,y_m) \neq 0$



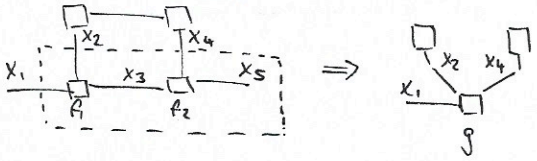
$$p(x_1,..,x_c, z_1,..,z_n \mid y_1,..,y_m) = p(x_1,..,x_c \mid y_1,..,y_m)\, p(z_1,..,z_n \mid y_1,..,y_m)$$

In other words, conditioned on $Y_1=y_1,..$ any subset of $x_1,..,x_c$ is independent of any subset of $z_1,..,z_n$.

Consequence of Thm 8.3: If prob. fun $p(x_{..})$ of some r.v. $x_{..}$ factors as $p(x_0,..,x_n) \propto g_0(x_0) \prod_{k=1}^{n} g_k(x_{k-1}, x_k)$ Then $x_{0..}$ form a Markov chain. Moreover for a hidden Markov model with fixed observations $Y_1=y_1,..$ the posterior distribution $p(x_0,..,x_n \mid y_1,..,y_n)$ is also a Markov chain.

8.4 Closing Boxes

Usually, and by default, close a box by summing or integrating over its internal variables. The resulting closed-box function is called external function of the box.



$$g(x_1,x_2,x_4) = \sum_{x_3}\sum_{x_5} f_1(x_1,x_2,x_3)\, f_2(x_3,x_4,x_5)$$

↑ Sum over all possible values of x_3 / x_5

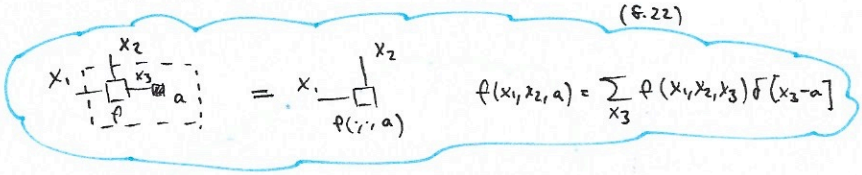
Thm 8.4 Closing an inner box within some outer box doesn't change the external function of the outer box (duh..!)

Partition Sum (PS)  "closing the entire graph" = The sum of the global function over the whole configuration space.

Closing a box in the FG does not change the partition sum.

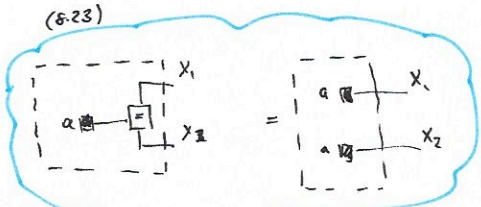
- if all factors are $\{0,1\}$-valued, the PS equals the number of valid configurations
- if a FG represents a prob. mass fun, its PS = 1
- if a FG represents $p(y_1,..,y_n, x_0,..,x_m)$ y: observables x: auxiliary, then the prob. of a particular observation $Y_1=y_1,..$ is $p(y_1,..,y_n) = \sum_{x_0} .. \sum_{x_m} p(y_1,..,y_n, x_0,..,x_m)$ which is the PS of the FG with the y_n plugged into the factors

(8.22)



$$f(x_1,x_2,a) = \sum_{x_3} f(x_1,x_2,x_3)\,\delta[x_3-a]$$

(8.23)



8.5 Matrix Multiplication and Such

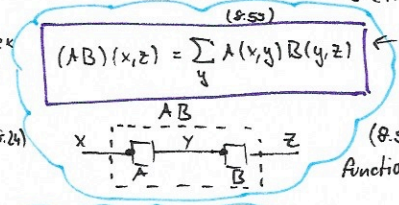
Closing boxes can be viewed as matrix multiplication.
Opening boxes ——— " ——— matrix factorization

$\mathbb{R}=\mathbb{R}/\mathbb{C}/\mathbb{Z}$ → matrix $A \in \mathbb{R}^{m\times n}$

mtx may be viewed as $\{1,..,m\}\times\{1,..,n\} \to R : (x,y) \mapsto A(x,y)$ x: row index, y: col index

$$(AB)(x,z) = \sum_y A(x,y)\,B(y,z)$$ ← mtx multiplication

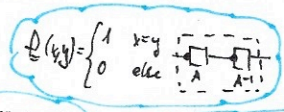
(8.55)



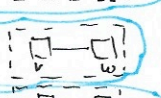
→ denotes row index

(8.55) is external function of fig 8.24

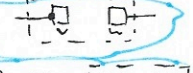
The identity mtx corresponds to equality check fun $\delta(x,y) = \begin{cases} 1 & x=y \\ 0 & \text{else} \end{cases}$ (8.24)

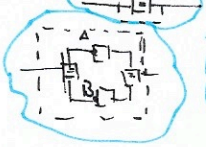


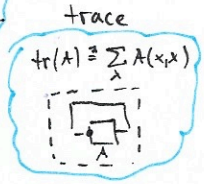
For vector $v \in \mathbb{R}^n$ $\{1,..,n\} \to R$ mtx product vA 

Inner product $\sum_x v(x)\,w(x)$ 

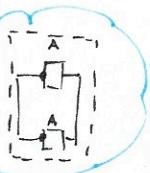
Diagonal mtx with diag vect v $v \cdot I$ 

Outer product vw^T 

Element wise multiplication (Hadamard product) 

trace $\mathrm{tr}(A) \triangleq \sum_x A(x,x)$ 

squared vector norm $\|v\|^2 = v^T v$ 

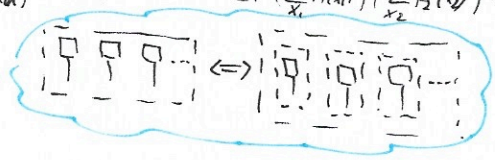
squared Frobenius norm $\|A\|^2 \triangleq \sum_{k=1}^{m}\sum_{l=1}^{n} (A_{k,l})^2$ 

8.6 Sum-Product Message Passing (Belief Propagation)

8.6.1 Problem Partition sum $Z_f \triangleq \sum_{x_1} \cdots \sum_{x_n} f(x_1,\dots,x_n)$ (8.61) or marginal $f_{x_n}(x_n) \triangleq \sum_{x_1,\dots,x_n \text{ except } x_n} f(x_1,\dots,x_n)$ (8.62) of many vars $x_1,\dots,x_n$.

→ Complexity exponential in n! Eg if $x_i \in \{0,1\}$ (8.61) has 2^n, (8.62) 2^{n-1} terms.

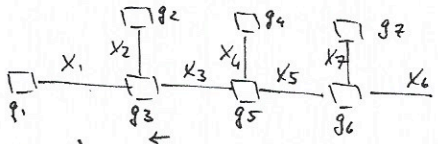
BUT if f can be factorized as $f(x_1,\dots,x_n) = f_1(x_1)\cdots f_n(x_n)$ (8.64) → $Z_f = \left(\sum_{x_1} f_1(x_1)\right)\left(\sum_{x_2} f_2(x_2)\right)\cdots$ n sums, $n-1$ multiplic. } linear $O(n)$



The following algorithm applies to FG w/o cycles which then allows computing (8.61) & (8.62) with linear complexity in n

8.6.2 The Algorithm Running example: $f_{x_3}(x_3) \triangleq \sum_{x_1,\dots,x_7 \text{ exp. } x_3} f(x_1,\dots,x_7)$

$f(x_1,\dots,x_7) = g_1(x_1)\, g_2(x_2)\, g_3(x_1,x_2,x_3)\cdots$

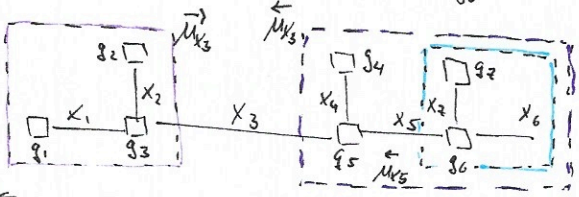


(I) Recursively close boxes

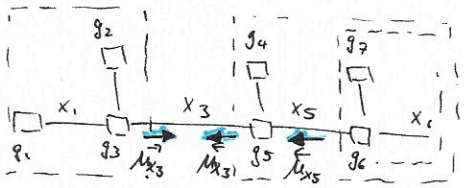
$$f_3(x_3) = \underbrace{\left(\sum_{x_1,x_2} g_1(x_1) g_2(x_2) g_3(x_1,x_2,x_3)\right)}_{\vec{\mu}_{x_3}} \cdot \left(\sum_{x_4,x_5} g_4(x_4) g_5(x_3,x_4,x_5)\left(\underbrace{\sum_{x_6,x_7} g_6(x_5,x_6,x_7) g_7(x_7)}_{\overleftarrow{\mu}_{x_5}}\right)\right) = \vec{\mu}_{x_3}(x_3)\,\overleftarrow{\mu}_{x_3}(x_3)$$

($\vec{\mu}_{x_3}$ under the first factor, $\overleftarrow{\mu}_{x_3}$ under the whole second factor)

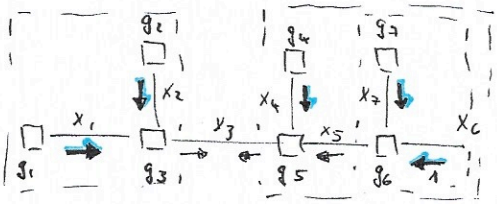
↑ corresponding closed-box functions



(II) Messages View the closed-box functions $\vec{\mu}$ as messages
These msg. are passed left to right $\vec{\mu}_{x_3}$ or right to left $\overleftarrow{\mu}_{x_3}$
They are functions, e.g. $\vec{\mu}_{x_3}$, $\overleftarrow{\mu}_{x_3}$ are fcn of x_3



(III) Messages out of leaf nodes These are simply the corresponding factors themselves: $\vec{\mu}_{x_1}(x_1) = g_1(x_1)$
Half-edges: $\overleftarrow{\mu}_{x_6}(x_6) = 1 \quad \forall x_6$

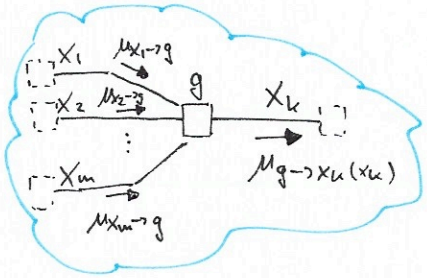


Sum-Product Message Computation Rule

The msg out of a leaf node along edge x is the function $g(x)$ itself.
The msg out of a non-leaf node along edge x_k is the function

$$\mu_{g\to x_k}(x_k) \triangleq \sum_{x_1,\dots,x_m \text{ exc. } x_k} g(x_1,\dots,x_m)\underbrace{\mu_{x_1\to g}(x_1)\cdots\mu_{x_m\to g}(x_m)}_{\text{except } \mu_{x_k\to g}(x_k)} \qquad (8.69a)$$

($\mu_{g\to x_k}(x_k) = \vec{\mu}_{x_k}$)



Thm 8.5 (Sum-Product Marginals) Let $f(x_1,\dots,x_n)$ be represented by a cycle-free FG and all messages are computed according to (8.69), then

$$\sum_{x_1,\dots,x_n \text{ except } x_k} f(x_1,\dots,x_n) = \vec{\mu}_{x_k}(x_k)\,\overleftarrow{\mu}_{x_k}(x_k) \qquad (8.70)$$

where $\overleftarrow{\mu}_{x_k}/\vec{\mu}_{x_k}$ are the two messages along the edge x_k.

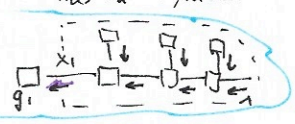
(this thm is just a consequence of thm 8.4)

8.6.3 Application to Partition Sums

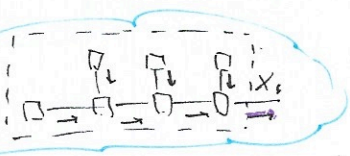
From a marginal function $f_{x_k}(x_k) \triangleq \sum_{x_1,\dots,x_n \text{ except } x_k} f(x_1,\dots,x_n)$ the partition sum is $Z_f = \sum_{x_k} f_{x_k}(x_k)$ (8.71)

In a cycle-free FG thus $Z_f = \sum_{x_k} \vec{\mu}_{x_k}(x_k)\,\overleftarrow{\mu}_{x_k}(x_k)$

$Z_f = \sum_{x_1} g_1(x_1)\,\overleftarrow{\mu}_{x_1}(x_1)$



$Z_f = \sum_{x_6} \vec{\mu}_{x_6}(x_6)$

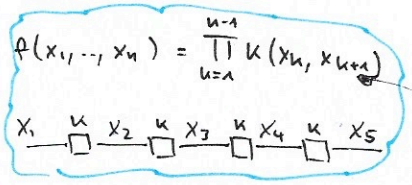


Ex 8.6 (Counting unconsecutive binary sequences)

Sequence $x_1, \dots, x_n$, $x_n \in \{0,1\}$. $f: \{0,1\}^n \to \{0,1\}$ (constraint function)

$$f(x_1, \dots, x_n) = \prod_{k=1}^{n-1} u(x_k, x_{k+1})$$

$$u(x_k, x_{k+1}) = \begin{cases} 0 & x_k = x_{k+1} = 1 \\ 1 & \text{otherwise} \end{cases}$$



FG of $n=5$

0101 ✓ → 2 : 0110, 1100, 1111, 0011 ✗

→ $f(x_1, \dots, x_n)$ is only $=1$ if the sequence $x_1, \dots, x_n$ has no consecutive ones'.

The desired number of valid sequences is the number of valid configurations → = the partition sum Z_f

eg. $Z_f = \sum_{x_n} \vec{\mu}_{x_n}(x_n) = \vec{\mu}_{x_n}(0) + \vec{\mu}_{x_n}(1)$. $\vec{\mu}_{x_1}(x_1) = 1$, $\vec{\mu}_{x_e}(x_e) = \sum_{x_{e-1}} u(x_{e-1}, x_e)\vec{\mu}_{x_{e-1}}(x_{e-1})$ ← by (8.69)

$$\vec{\mu}_{x_2}(x_2) = u(0,x_2) + u(1,x_2) = \begin{cases} 2 & x_2 = 0 \\ 1 & x_2 = 1 \end{cases} \qquad \vec{\mu}_{x_3}(x_3) = u(0,x_3)\vec{\mu}_{x_2}(0) + u(1,x_3)\vec{\mu}_{x_2}(1) = \begin{cases} 2+1 & x_3=0 \\ 2+0 & x_3=1 \end{cases} = \begin{cases} 3 & x_3=0 \\ 2 & x_3=1 \end{cases}$$

$$\vec{\mu}_{x_4}(x_4) = u(0,x_4)\vec{\mu}_{x_3}(0) + u(1,x_4)\vec{\mu}_{x_3}(1) = \begin{cases} 5 & x_4=1 \\ 3 & x_4=0 \end{cases} \qquad \vec{\mu}_{x_5}(x_5) = \begin{cases} 8 & x_5=1 \\ 5 & x_5=0 \end{cases}$$

$$Z_f = \sum_{x_5} \vec{\mu}_{x_5}(x_5) = 8+5 = 13 \quad ✓$$

8.6.4 Application to Marginal Probabilities

Let $p(x_1, \dots, x_n)$ be a joint prob. density represented by a cycle-free FG.

Then $p(x_n) = \sum_{x_1, \dots, x_n \text{ except } x_n} p(x_1, \dots, x_n) = \vec{\mu}_{x_n}(x_n)\overleftarrow{\mu}_{x_n}(x_n)$

can be directly computed in FG of f and then recover γ^{-1} from $\sum p(x_n) = 1$

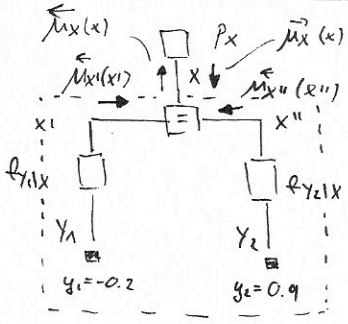
More generally a FG represents $f(x_1, \dots, x_n) = \gamma^{-1} p(x_1, \dots, x_n)$ $\gamma \neq 0$, $\gamma^{-1} = Z_f$ $\gamma^{-1} p(x_n) = \vec{\mu}_{x_n}(x_n)\overleftarrow{\mu}_{x_n}(x_n)$

This applies to conditional probs: (8.84)

$$p(x_n \mid y_1, \dots, y_m) = \sum_{\substack{x_1, \dots, x_n \\ \text{exc. } x_n}} p(x_1, \dots, x_n \mid y_1, \dots, y_m) \propto \sum_{\substack{x_1, \dots, x_n \\ \text{exc. } x_n}} p(x_1, \dots, x_n, y_1, \dots, y_m)$$

Ex 8.8 (Noisy Indep. Measurements)

$X \in \{0,1\}$, Y_1, Y_2 indep. meas of X $f(y_k \mid x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k - x)^2}{2\sigma^2}}$ $k=1,2$ $\sigma = 1$



Assume $P_X(1) = 0.1$, $Y_1 = y_1 = -0.2$ $Y_2 = y_2 = 0.9$, wish to compute

$$p(x \mid y_1, y_2) = \frac{p(x) f(y_1, y_2 \mid x)}{f(y_1, y_2)} \propto p(x) f(y_1, y_2 \mid x) = \sum_{x', x''} p(x)\delta[x - x']\delta[x - x'']\, f_{Y_1|X}(y_1 \mid x')\, f_{Y_2|X}(y_2 \mid x'')$$

By sum-prod. mess. passing: $\vec{\mu}_X(x) = p(x)$

$$\overleftarrow{\mu}_{x_1}(x') = f_{Y_1|X}(-0.2 \mid x') = \begin{pmatrix} 0 \mapsto 0.391 \\ 1 \mapsto 0.194 \end{pmatrix} \qquad \overleftarrow{\mu}_{x''}(x'') = f_{Y_2|X}(0.9 \mid x'') = \begin{pmatrix} 0 \mapsto 0.266 \\ 1 \mapsto 0.397 \end{pmatrix}$$

Close box: $\overleftarrow{\mu}_X(x) = \sum_{x', x''} \overleftarrow{\mu}_{x_1}(x')\overleftarrow{\mu}_{x''}(x'')\delta[x - x']\delta[x - x''] = \overleftarrow{\mu}_{x_1}(x)\overleftarrow{\mu}_{x''}(x) = \begin{pmatrix} 0 \mapsto 0.104 \\ 1 \mapsto 0.077 \end{pmatrix}$

Finally $p(x \mid y_1, y_2) \propto \vec{\mu}_X(x)\overleftarrow{\mu}_X(x) = \begin{pmatrix} 0 \mapsto 0.9 \cdot 0.104 \\ 1 \mapsto 0.1 \cdot 0.077 \end{pmatrix}$, by normalization $p(x \mid y_1, y_2) = \begin{pmatrix} 0 \mapsto 0.924 \\ 1 \mapsto 0.076 \end{pmatrix}$

8.7 Closing Boxes by Maximization & Max-Product Message Passing

Instead of closing boxes by sum/integrate we can do so by maximizing over all internal variables

Thm 8.6 Assuming all nodes/factors are real and nonnegative, closing an inner box within an outer box by maximizing over its internal variables doesn't change the external function of the outer box.

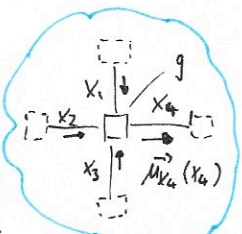
It follows that message passing still holds when replacing summation with maximization

Max-Product Message Computation Rule

$$\mu_{g \to x_k}(x_k) = \vec{\mu}_{x_k}(x_k) = \begin{cases} g(x_k) & \text{if leaf} \\ \max \cdots & \text{else} \end{cases}$$

(8.107)

$$\mu_{g \to x_k}(x_k) \triangleq \max_{x_1, \dots, x_m \text{ except } x_k} g(x_1, \dots, x_m)\underbrace{\mu_{x_1 \to g}(x_1) \cdots \mu_{x_m \to g}(x_m)}_{\text{except } \mu_{x_k \to g}(x_k)}$$



Thm 8.7 (Max-Product Marginals)

Let $f(x_1, \dots, x_n)$ represented by cycle-free FG with real and nonnegative factors/nodes then

(8.102)

$$\max_{\substack{x_1, \dots, x_n \\ \text{except } x_k}} f(x_1, \dots, x_n) = \vec{\mu}_{x_k}(x_k)\overleftarrow{\mu}_{x_k}(x_k)$$

Importance:

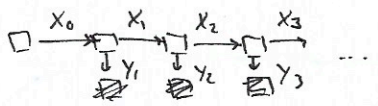
$$(\hat{x}_1, \dots, \hat{x}_n) \triangleq \underset{x_1, \dots, x_n}{\operatorname{argmax}} f(x_1, \dots, x_n)$$

$$\hat{x}_k = \underset{x_k}{\operatorname{argmax}}\, \vec{\mu}_{x_k}(x_k)\overleftarrow{\mu}_{x_k}(x_k)$$

8.8 Message Passing in Markov Chains and Hidden Markov Models (HMM)

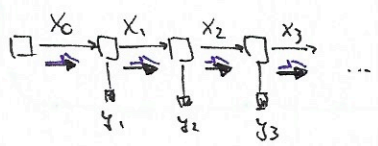
8.8.1 Msg Passing in HMM

Recall characterization of HMM: $p(x_0,\dots,x_n,y_1,\dots,y_n) = p(x_0)\prod_{k=1}^{n} p(x_k,y_k|x_{k-1})$



Current state estimation

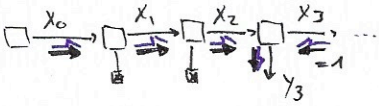
$$p(x_n|y_1,\dots,y_n) \propto \sum_{x_0}\cdots\sum_{x_{n-1}} p(x_0,\dots,x_n,y_1,\dots,y_n) = \vec{\mu}_{x_n}(x_n)$$



can be calculated by forward sum-prod. msg passing

Prediction of next output:

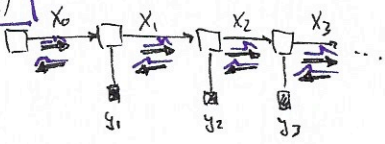
$$p(y_n|y_1,\dots,y_{n-1}) \propto \sum_{x_0,\dots,x_n} p(x_0,\dots,x_n,y_1,\dots,y_n) = \vec{\mu}_{y_n}(y_n)$$



forward sum-prod. msg passing

Estimation of state at any time:

$$p(x_n|y_1,\dots,y_n) \propto \sum_{\substack{x_0,\dots,x_n \\ \text{except } x_n}} p(x_0,\dots,x_n,y_1,\dots,y_n) = \vec{\mu}_{x_n}(x_n)\overleftarrow{\mu}_{x_n}(x_n)$$



forward-backward sum-prod. msg. passing

MAP estimate of state trajectory:

$$(\hat{x}_0,\dots,\hat{x}_n)_{MAP} \triangleq \underset{x_0,\dots,x_n}{\operatorname{argmax}}\, p(x_0,\dots,x_n|y_1,\dots,y_n)$$ if is unique

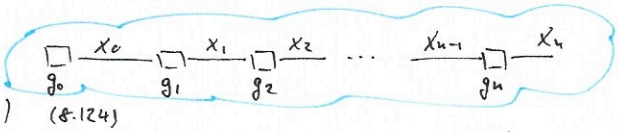
$$\rightarrow \hat{x}_k = \underset{x_k}{\operatorname{argmax}}\, \vec{\mu}_{x_k}(x_k)\overleftarrow{\mu}_{x_k}(x_k)$$

Probability of observation

$$p(y_1,\dots,y_n) = \sum_{x_0,\dots,x_n} p(x_0,\dots,x_n,y_1,\dots,y_n) = \sum_{x_n}\sum_{x_0,\dots,x_{n-1}} p(x_0,\dots,x_n,y_1,\dots,y_n) = \sum_{x_n} \vec{\mu}_{x_n}(x_n)$$

[Scale factors in $\vec{\mu}_x$ cannot be ignored in this case!]

8.8.2 MC Sampling and Reparametrization



If $x_0,\dots,x_n$ is a MC $p(x_0,\dots,x_n) = p(x_0)p(x_1|x_0)p(x_2|x_1)\cdots p(x_n|x_{n-1})$ (8.124)

Creating a sample $(x_0,\dots,x_n)$ from (8.124) (i.e. simulating the MC) is straightforward:

I. draw x_0 from $p(x_0)$ II. draw x_1 from $p(x_1|x_0)$ III. etc... (8.125)

Sampling is less obvious, but if $p(x_0,\dots,x_n)$ is given by $\propto g_0(x_0)\prod_{k=1}^{n} g_k(x_{k-1},x_k)$ with general factors g_k

Sampling problems

Ex 8.9 (Noncoincidin sim. seq.): Wish to generate rnd seq $(x_0,\dots,x_n) \in \{0,1\}^{n+1}$ w/o coinciding ones, uniformly dist. over all such seq. this is a MC with $g_0(x_0)=1$ and $g_k(x_{k-1},x_k) = \begin{cases} 0 & \text{if } x_{k-1}=x_k=1 \\ 1 & \text{else} \end{cases}$

Ex 8.11 (Sampling from MC with fixed final state): $X_0X_1,\dots X_n$ a MC with fixed $X_n=x_n$. Wish to find $(x_0,\dots,x_n)$ according to $p(x_0,\dots,x_{n-1}|x_n)$.

Since (8.125) is MC can be parametrized as (8.124) s.t. $p(x_0) = g_0(x_0)\overleftarrow{\mu}_{x_0}(x_0)$

and $$p(x_k|x_{k-1}) = \frac{p(x_{k-1},x_k)}{p(x_{k-1})} = \frac{\vec{\mu}_{x_{k-1}}(x_{k-1})\,g_k(x_{k-1},x_k)\overleftarrow{\mu}_{x_k}(x_k)}{\vec{\mu}_{x_{k-1}}(x_{k-1})\overleftarrow{\mu}_{x_{k-1}}(x_{k-1})} = \frac{g_k(x_{k-1},x_k)\overleftarrow{\mu}_{x_k}(x_k)}{\overleftarrow{\mu}_{x_{k-1}}(x_{k-1})}$$ (8.130)

↑ sum-product

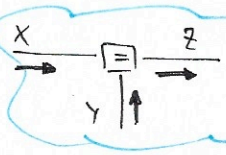
Sampling can thus be done by backwards sum-prod. msg passing followed by forward sampling.

8.9 Additional Remarks and Details

Message Scaling $\vec{\mu}(\cdot)$ typically tend to quickly zero / infty. In practice they are scaled / normalized. May have to keep track of scale factor, in log form.

Message Representation $\vec{\mu}(\cdot)$ are functions. If domain is finite, store tables. If domain is binary, store $\frac{\vec{\mu}(0)-\vec{\mu}(1)}{\vec{\mu}(0)+\vec{\mu}(1)}$ or $\log \frac{\vec{\mu}(0)}{\vec{\mu}(1)}$

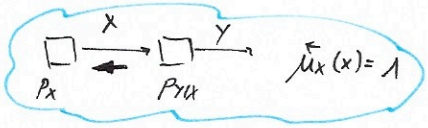
Msg. passing through eq. constraints



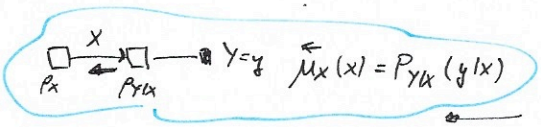
$$\vec{\mu}_Z(z) = \sum_x \sum_y f_=(x,y,z)\vec{\mu}_X(x)\vec{\mu}_Y(y) = \vec{\mu}_X(z)\vec{\mu}_Y(z)$$

or $\vec{\mu}_Z(z) = \max \cdots = \vec{\mu}_X(z)\vec{\mu}_Y(z)$ for max-prod.

Backward Msg in Chain-Rule Models FG of $p(x,y) = p(x)p(y|x)$



Y unknown



← likelihood function

Y known

Message passing in FG with cycles

- Result can be arbitrarily bad
- Is used with excellent rates, e.g. in decoding error correcting codes

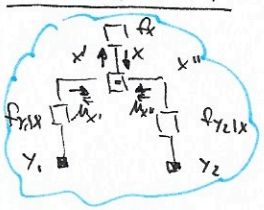
- Initialize all $\vec{\mu} = \overleftarrow{\mu} = 1$
- Iterate thru msgs with some schedule
- Stop after time / condition

Chapter 10: Gaussian Message Passing

10.1 Introduction

Notes:
- Generalization of sum-prod. msg passing to continuous variables/Rvs. is by replacing sum by integral
- $\delta[\cdot] \to \delta(\cdot)$

Ex10.1 (cf. Ex8.8)



Given: $X \in \mathbb{R}$ r.v. $f(x)$, $Y_1, Y_2 \in \mathbb{R}$ observations with $f(y_1,y_2|x) = f(y_1|x)f(y_2|x)$ $Y_1 = y_1$, $Y_2 = y_2$

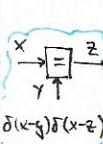
Problem: Find posterior $f(x|y_1,y_2)$

Sol: $f(x|y_1,y_2) \propto \vec{\mu}_X(x)\overleftarrow{\mu}_X(x)$ $\vec{\mu}_X(x) = f(x)$ $\overleftarrow{\mu}_{X'}(x') = f_{Y_1|X}(y_1|x')$ $\overleftarrow{\mu}_{X''}(x'') = f_{Y_2|X}(y_2|x'')$

$$\overleftarrow{\mu}_X(x) = \int_{x'}\int_{x''} \overleftarrow{\mu}_{X'}(x')\overleftarrow{\mu}_{X''}(x'')\delta(x-x')\delta(x-x'')dx'dx'' = f_{Y_1|X}(y_1|x)\, f_{Y_2|X}(y_2|x)$$

Focus lies now on scalar Gaussian messages described by mean m and variance σ^2

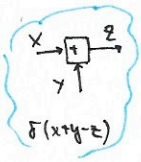
$$\mu(x) \propto \exp\left(\frac{-(x-m)^2}{2\sigma^2}\right) \propto \exp\left(-x^2\frac{1}{2\sigma^2} + x\frac{m}{\sigma^2}\right)$$



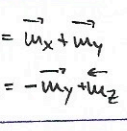
$$\vec{m}_Z = \frac{\vec{m}_X/\vec{\sigma}_X^2 + \vec{m}_Y/\vec{\sigma}_Y^2}{1/\vec{\sigma}_X^2 + 1/\vec{\sigma}_Y^2} \quad (10.11)$$

$$1/\vec{\sigma}_Z^2 = 1/\vec{\sigma}_X^2 + 1/\vec{\sigma}_Y^2 \quad (10.12)$$

$$\overleftarrow{m}_X = \frac{\vec{m}_Y/\vec{\sigma}_Y^2 + \overleftarrow{m}_Z/\overleftarrow{\sigma}_Z^2}{1/\vec{\sigma}_Y^2 + 1/\overleftarrow{\sigma}_Z^2} \qquad 1/\overleftarrow{\sigma}_X^2 = 1/\vec{\sigma}_Y^2 + 1/\overleftarrow{\sigma}_Z^2$$



$$\vec{m}_Z = \vec{m}_X + \vec{m}_Y \qquad \vec{\sigma}_Z^2 = \vec{\sigma}_X^2 + \vec{\sigma}_Y^2$$
$$\overleftarrow{m}_X = -\vec{m}_Y + \overleftarrow{m}_Z \qquad \overleftarrow{\sigma}_X^2 = \vec{\sigma}_Y^2 + \overleftarrow{\sigma}_Z^2$$



$$\vec{m}_Y = a\vec{m}_X \qquad \vec{\sigma}_Y^2 = a^2\vec{\sigma}_X^2$$
$$\overleftarrow{m}_X = \overleftarrow{m}_Y/a \qquad \overleftarrow{\sigma}_X^2 = \overleftarrow{\sigma}_Y^2/a^2$$

Incoming Gaussian messages to these nodes are also Gaussian at their outputs.

Note: The means never affect the variances (but not vice-versa)

Ex10.2 (Ex10.1 contd) Let $f(x) = \mathcal{N}(m_x, \sigma_x^2)$ $f(y_k|x) = \mathcal{N}(x, \sigma^2)$ We then have $\vec{m}_X = m_X$ $\vec{\sigma}_X^2 = \sigma_X^2$ $\overleftarrow{m}_{X'} = y_1$ $\overleftarrow{\sigma}_{X'}^2 = \sigma^2$

Then by (10.11)/(10.12) $\overleftarrow{m}_X = (y_1+y_2)/2$ $\overleftarrow{\sigma}_X^2 = \sigma^2/2$

$$m_{X|Y} = \frac{\vec{m}_X/\vec{\sigma}_X^2 + \overleftarrow{m}_X/\overleftarrow{\sigma}_X^2}{1/\vec{\sigma}_X^2 + 1/\overleftarrow{\sigma}_X^2} = \frac{m_X/\sigma_X^2 + (y_1+y_2)/\sigma^2}{1/\sigma_X^2 + 2/\sigma^2} \qquad 1/\sigma_{X|Y}^2 = 1/\sigma_X^2 + 2/\sigma^2$$

Remarks from Appendix C Facts about Gaussian vectors:

1. For Gaussian msg., sum-prod. coincides with max-prod. msg passing
2. For jointly Gaussian r.v.: MAP = MMSE = LMMSE estimates
3. Even if X, Y not jointly Gaussian, the LMMSE of X from $Y=y$ may be obtained by pretending X, Y jointly Gaussian and forming that correspondingly MAP

Merge:

$$m = \frac{\vec{m}/\vec{\sigma}^2 + \overleftarrow{m}/\overleftarrow{\sigma}^2}{1/\overleftarrow{\sigma}^2 + 1/\vec{\sigma}^2} \qquad \sigma^2 = \frac{1}{1/\vec{\sigma}^2 + 1/\overleftarrow{\sigma}^2}$$

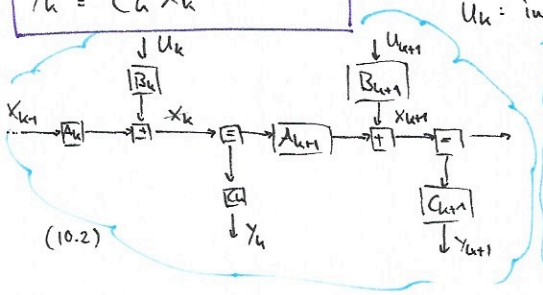
10.2 Linear Gaussian State Space Models

$$x_k = A_k x_{k-1} + B_k u_k$$
$$y_k = C_k x_k$$

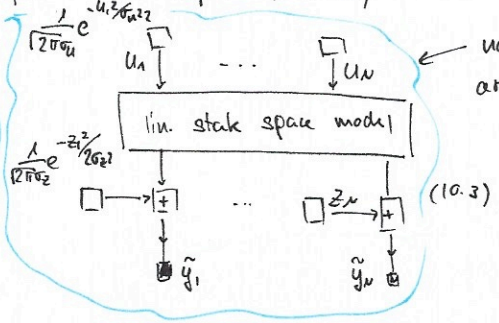
Linear state space model:

u_k, x_k, y_k real col. vectors A_k, B_k, C_k real matrices

u_k: input y_k: output, mostly scalar x_k: state vector



(10.2)



(10.3)

← noisy measurements and inputs are Gaussian

The initial state x_0 and input signal $u_1, .., u_N$ fully determine both the state trajectory $x_1, .., x_N$ and the noise free output $y_1, .., y_N$. In consequence, the MAP/MMSE/LMMSE of any of these quantities is fully determined by the minimizer of

$$\underset{x_0, u_1, .., u_N}{\operatorname{argmax}} \prod_{k=1}^{N} e^{-u_k^2/2\sigma_u^2} e^{-(y_k - \tilde{y}_k)^2/2\sigma_z^2} = \underset{x_0, u_1, .., u_N}{\operatorname{argmin}} \left(\frac{1}{\sigma_u^2} \sum_{k=1}^{N} u_k^2 + \frac{1}{\sigma_z^2} \sum_{k=1}^{N} (y_k - \tilde{y}_k)^2 \right) \quad (10.33)$$

10.3 Vector-Gaussian Message Passing and Kalman Filtering

Gaussian Message Passing through graph (10.3) with (10.2) plugged in is known as Kalman Filtering!

Recall: vector $\vec{x}$ of Gaussian r.v. (real)

$$f_{\vec{x}}(\vec{x}) \propto e^{-\frac{1}{2}(\vec{x}-\vec{m})^T \vec{V}^{-1} (\vec{x}-\vec{m})}$$

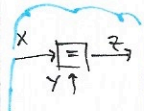
$\vec{m}$: mean vector

$\vec{V}$: covariance matrix

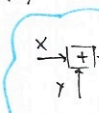
$\vec{W} = \vec{V}^{-1}$: precision matrix

$\vec{\xi} \triangleq \vec{W}\vec{m}$: some vector

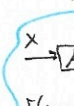
exact: $f_{\vec{x}}(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n}} \cdot \frac{1}{\sqrt{\det(V)}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{m})^T \vec{V}^{-1} (\vec{x}-\vec{m})\right)$



$$\vec{W}_Z = \vec{W}_X + \vec{W}_Y$$
$$\vec{\xi}_Z = \vec{\xi}_X + \vec{\xi}_Y$$
$$\overleftarrow{W}_X = \overleftarrow{W}_Z + \overleftarrow{W}_Y$$
$$\overleftarrow{\xi}_X = \overleftarrow{\xi}_Z + \overleftarrow{\xi}_Y$$



$$\vec{V}_Z = \vec{V}_X + \vec{V}_Y$$
$$\vec{m}_Z = \vec{m}_X + \vec{m}_Y$$
$$\overleftarrow{V}_X = \overleftarrow{V}_Z + \overleftarrow{V}_Y$$
$$\overleftarrow{m}_X = \overleftarrow{m}_Z - \overleftarrow{m}_Y$$



$Y = AX$

real, and rank

$$\vec{V}_Y = A \vec{V}_X A^T$$
$$\vec{m}_Y = A \vec{m}_X$$
$$\overleftarrow{W}_X = A^T \overleftarrow{W}_Y A$$
$$\overleftarrow{\xi}_X = A^T \overleftarrow{\xi}_Y$$

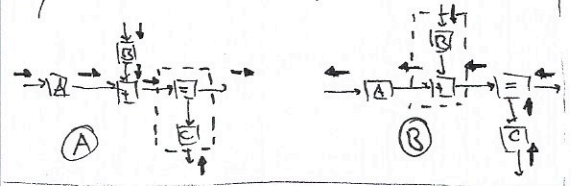
$$\vec{V}_Z = \vec{V}_X - \vec{V}_X A^T G A \vec{V}_X$$
$$G \triangleq \left(\overleftarrow{V}_Y + A \vec{V}_X A^T\right)^{-1}$$
$$\vec{m}_Z = \vec{m}_X + \vec{V}_X A^T G (\overleftarrow{m}_Y - A \vec{m}_X)$$

(A)

$$\vec{W}_Z = \vec{W}_X - \vec{W}_X A H A^T \vec{W}_X$$
$$H \triangleq \left(\vec{W}_Y + A^T \vec{W}_X A\right)^{-1}$$
$$\vec{\xi}_Z = \vec{\xi}_X + \vec{W}_X A H \left(\overleftarrow{\xi}_Y - A^T \vec{\xi}_X\right)$$
$$\overleftarrow{W}_X = \overleftarrow{W}_Z - \overleftarrow{W}_Z A K A^T \overleftarrow{W}_Z$$
$$K \triangleq \left(\vec{W}_Y + A^T \overleftarrow{W}_Z A\right)^{-1}$$
$$\overleftarrow{\xi}_X = \ldots$$

(B) Thm 3

Thm 10.3 is used to omit the many matrix inversions. In this way, if both u_k, y_k are scalars both fwd and bwd msg. passing is accomplished without any matrix inversion.



Remarks: (1.) With these tables we can translate computations of section 8.8 (MC, HMM) to lin. Gaussian SS models. In particular, can compute MAP, MMSE- of current, past states and next output.

(2.) The final (MAP,..) estimate of some quantity x is the mean vector m_x of the scaled Gaussian $\mu_x(x) \triangleq \vec{\mu}_x(x)\overleftarrow{\mu}_x(x)$

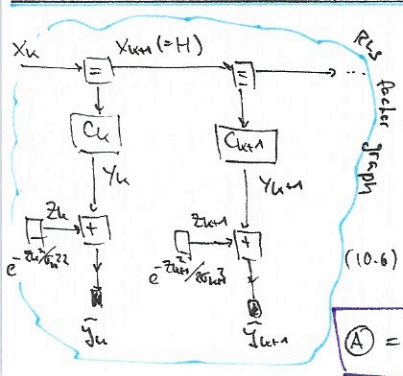
(3.) A known variable s.a. $\tilde{y}_k = \tilde{y}_k$ leads to degenerate Gaussian $\overleftarrow{\mu}_{\tilde{y}_k}$, $\overleftarrow{m}_{\tilde{y}_k} = \tilde{y}_k$ $\overleftarrow{V}_{\tilde{y}_k} = 0$. If not observed $\overleftarrow{m}_{\tilde{y}_k} = 0$ $\overleftarrow{W}_{\tilde{y}_k} = 0$

(4.) Comp. rules for $\vec{V}, \vec{W}$ do not involve any mean vectors → can be precomputed before $\tilde{y}_k$ is observed.

Applies in particular in stationary when A_k, B_k, C_k do not depend on k

(5.) Numerical problems will arise. With $A_k = A$ in Jordan form often helps

10.5 Recursive Least Squares (RLS) and Adaptive Filters



(10.6)

(10.6) may be viewed as (10.2)/(10.3) with $A_k = I$ $B_k = 0$. x_k: unknown real-valued col. vector, $c_1, c_2, ..$ real valued ~~row~~ vectors. The unknown $x_k = x_{k+1} = .. = H$

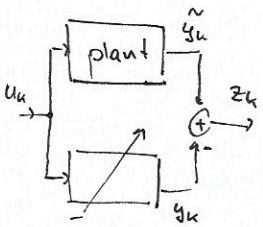
(10.6) expresses $\tilde{y}_k = c_k H + z_k$, $z_1, ..$ real, zero-mean, indep. Gaussian.

10.5.1 RLS Recall LS-problem $\|Ax - \tilde{y}\|$ find x. Let c_k be k-th row of A, $\tilde{y}_k$ k-th component of $\tilde{y}$. Then $\|Ax - \tilde{y}\|^2 = \sum_k (c_k x - \tilde{y}_k)^2$ thus

$$\underset{x}{\operatorname{argmin}} \|Ax - \tilde{y}\| = \underset{x}{\operatorname{argmax}} e^{-\|Ax-\tilde{y}\|^2} = \underset{x}{\operatorname{argmax}} \prod_k e^{-(c_k x - \tilde{y}_k)^2}$$

(A) $= \prod_k e^{-z_k^2}$ is represented by Fig 10.6 → $\hat{x} = \hat{x}$ may be found by max-(=sum-)prod. Gaussian msg. passing

10.5.2 Adaptive FIR Filters



Given: real input $u_{1-M}, \dots, u_0, u_1, u_2, \dots, u_N$
desired real output $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N$

Whish: find filter taps $h_0, \dots, h_M$ s.t. $y_k = \sum_{\ell=0}^{M} h_\ell u_{k-\ell}$ is a good approx. of $\tilde{y}_k$

Specifically $\sum_{k=1}^{N} (\tilde{y}_k - y_k)^2$ as small as possible.

With $h \triangleq (h_0, h_1, \dots h_M)^T$ $c_k \triangleq (u_k, u_{k-1}, \dots, u_{k-M})$ $y_k = \sum h_\ell u_{k-\ell} = c_k h$

and the desired sol.: $\hat{h} = \operatorname{argmin}_h \sum_k (\tilde{y}_k - c_k h)^2 = \operatorname{argmax}_h \prod_k e^{-(\tilde{y}_k - c_k h)^2}$ which is again the maximization of FG 10.6

10.5.4 RLS Algorithms as Message Passing

- Actual sum-prod. = max-prod. msg. passing through 10.6 is equivalent to classic RLS algos.
- Forward-only (left to right) is enough: If the last observation is $\tilde{y}_N$, the desired sol. $\hat{h}$ is the mean vector $\vec{m}_{X_{N+1}}$ of the message $\vec{\mu}_{X_{N+1}}$.
- Two methods: I. Propagate $\vec{W}_{X_k}, \vec{S}_{X_k}$, extract mean vector by solving $\vec{S}_{X_k} = \vec{W}_{X_k} \vec{m}_{X_k}$
  II. Propagate $\vec{V}_{X_k}, \vec{m}_{X_k}$, $\vec{m}_{X_k}$ is already available at all times
- In every step multiply $\vec{V}_{X_k}$ in each step by $\gamma > 1$, $\gamma \approx 1$ scalar to slowly forget the past and improve numerical stability.

Chapter 12: Monte Carlo Methods (MC)

A probability density can be approx. represented by a list of samples drawn from the density.
$x^{(1)}, \dots, x^{(L)}$ are (suff. indep.) samples from f_x and g a function defined on the range of x

$$E[g(x)] = \int_x g(x) f_x(x) dx \approx \frac{1}{L} \sum_{\ell=1}^{L} g(x^{(\ell)}) \quad (12.2)$$

with identity for $L \to \infty$ w.p.1

If g is indicator function of some set S $g(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$ then

$$P(x \in S) = \int_x g(x) f_x(x) dx \approx \frac{1}{L} \sum_{\ell=1}^{L} g(x^{(\ell)}) = \frac{1}{L} \sum_{x^{(\ell)} \in S} 1$$

Generally it is sometimes attractive to work with samples not from f_x but from some aux. dist q.

$$\hat{E}[g(x)] \triangleq \frac{1}{L} \sum_{\ell=1}^{L} \frac{f_x(x^{(\ell)})}{q(x^{(\ell)})} \cdot g(x^{(\ell)}) \quad (12.7)$$

$\hat{E}[g(x)]$ with respect to q

$E_q[\hat{E}[g(x)]] = E[g(x)]$ E respect to f_x

this is called importance sampling bcs the var of (12.7) may be larger/smaller than the one of (12.2)

Example: Samples $(x^{(1)}, y^{(1)}), \dots$ from $f_{x,y}$ can be turned into a list of samples from f_y by dropping the $x^{(\cdot)}$ component of each sample.

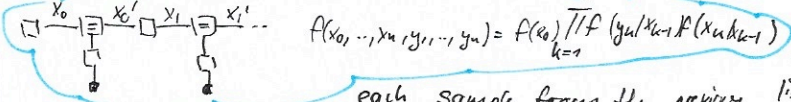
A fundamental limitation of MC is the prob. of very rare events cannot be assessed from a list of samples of practical size. the convergence of (12.2) is not guaranteed.

12.1 Particle Filter

= seq. MC algorithm

Problem: Implementation of sum-prod. msg. passing if not Gaussian (= continuous alphabet) is hard!
→ MC can help: lists of samples may be used to represent joint dist., messages and marginals.

Illustrate by application to state estimation: Fwd. sum-prod. msg passing with samples ("particles"):



$f(x_0, \dots, x_n, y_1, \dots, y_n) = f(x_0) \prod_{k=1}^{n} f(y_k | x_k) f(x_k | x_{k-1})$

1. init: list of indep. samples $[x_0^{(\ell)}]$ from f_{x_0} represent $\vec{\mu}_{X_0}$
2. For $k = 1, 2, \dots, n-1$ repeat steps 3-4
3. Create new list of indep. samples $[x_k'^{(\ell)}]$ by selecting each sample from the previous list $[x_k^{(\ell)}]$ (with repetitions) with probability proportional to $f_{Y_{k+1}|X_k}(y_{k+1} | x_k^{(\ell)})$. The new samples represent $\vec{\mu}_{X_k'}(x) = \vec{\mu}_{X_k}(x) f_{Y_{k+1}|X_k}(y_{k+1} | x)$
4. The msg. $\vec{\mu}_{X_{k+1}}$ is represented by samples $[x_{k+1}^{(\ell)}]$, which are sampled indep. from $f_{X_{k+1}|X_k}(\cdot | x_k'^{(\ell)})$

12.2 Gibbs Sampling

General method for creating samples from some joint pros. $p(x) = p(x_1, \dots, x_n)$

+ Can be used for FG with cycles
− Samples are not indep., dependence btw $x^{(\ell)}, x^{(\ell+m)}$ may decay slowly in m

Consider case $n=2$. ▽ Samples form a Markov Chain ▽

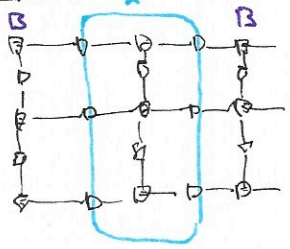
1. Begin with sample $x_2^{(0)}$ from $p(x_2)$
2. $x_1^{(\ell)}$ is sampled according to $p_{X_1|X_2}(x_1|x_2^{(\ell-1)})$ & $p_{X_1,X_2}(x_1, x_2^{(\ell-1)})$
3. $x_2^{(\ell)}$ ——— $p_{X_2|X_1}(x_2|x_1^{(\ell)})$ & $p_{X_1,X_2}(x_1^{(\ell)}, x_2)$

} repeat

Step 1. is difficult. Practice: Choose $x_2^{(0)}$ from other dist. or deterministic with $p_{X_2}(x_2^{(\cdot)}) > 0$
→ need some time to converge to p_{X_1,X_2}

For $n > 2$: initial $\to x_1^{(1)}, x_2^{(2)}$ with for $k=1$ to n {
sample $x_k^{(\ell)}$ from $q_k^{(\ell)}(x_k) \triangleq p_{X_k|X_1,\dots,X_{k-1},X_{k+1},\dots,X_n}(x_k|x_1^{(\ell)},\dots,x_{k-1}^{(\ell)},x_{k+1}^{(\ell)},\dots,x_n^{(\ell)})$
& $p_{X_1,\dots,X_n}(x_1^{(\ell)},\dots,x_{k-1}^{(\ell)},x_{k+1}^{(\ell)},\dots,x_n^{(\ell)})$

Improving Gibbs Sampling



Let $p(x_1,\dots,x_n)$ be represented by FG.
→ Partition FG into sets without cycles
- Apply Gibbs on block level by alternating between sampling from $p(x_A|x_B)$ with x_B fixed and $p(x_B|x_A)$ with x_A fixed. Both steps can be carried out by backward filtering, forward sampling.

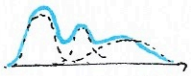
→ Example application on Restricted Boltzmann Machine

Chapter 13: Parameter Esti. w/ Expectation Maximization

Goal: Determine statistical model parameters
e.g. $\hat{\theta} = \arg\max_\theta p(y^{(1)},\dots,y^{(L)}|\theta)$ with data $y^{(\cdot)}$

13.1 Gaussian-Mixture Estimation (GM)

A scalar GM is a pros. density $f(y) = \sum_{k=1}^{K} p_k f_k(y)$ (13.2) $K>1$ $\sum p_k = 1$ $f_k(y) \sim \mathcal{N}(m_k, \sigma_k^2)$,



same as (13.2) (13.4)
$f(y) = \sum_{x=1}^{K} p(x) f(y|x)$ X: r.v. $\in \{1,\dots,K\}$, $f(y|x) = f_k(y)$

13.1.1 Problem

Given indep. samples $y^{(\cdot)} \in \mathbb{R}$ drawn from (13.2), find p_k, m_k, σ_k^2. $L \gg K$

Special case $K=1$: $\hat{m} = \frac{1}{L}\sum_{\ell}^{L} y^{(\ell)}$ (13.5), $\hat{\sigma}^2 = \frac{1}{L}\sum_{\ell}(y^{(\ell)} - \hat{m})^2 = \frac{1}{L}\sum_{\ell}(y^{(\ell)})^2 - \hat{m}^2$ (13.7) [1] or $\hat{\sigma}^2 = \frac{1}{L-1}\sum_{\ell=1}^{L}(y^{(\ell)} - \hat{m})^2 = \frac{1}{L-1}\sum(y^{(\ell)})^2 - \frac{1}{L-1}\hat{m}^2$ (13.9) [1]

[1] are ML estimates

13.1.2 Algorithm

For $K>1$ ∄ a closed form sol.

1. Initial guess $\hat{\theta} = (\hat{p}_1,\dots,\hat{p}_K, \hat{m}_1,\dots,\hat{m}_K, \hat{\sigma}_1^2,\dots,\hat{\sigma}_K^2)$
2. For $\ell=1,\dots,L$, $k=1,\dots,K$ compute posterior according to current model (13.10)
$$P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta}) = \frac{\hat{p}_k \hat{f}_k(y^{(\ell)})}{\sum_{k=1}^{K} \hat{p}_k \hat{f}_k(y^{(\ell)})}$$
$X^{(\ell)}$: indexes components as in (13.4)
$\hat{f}_k(y)$: $\sim \mathcal{N}(\hat{m}_k, \hat{\sigma}_k^2)$
3. Compute new estimates

$$\hat{p}_k = \frac{1}{L}\sum_{\ell=1}^{L} P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta}) \quad (13.11)$$

$$\hat{m}_k = \frac{1}{s_k}\sum_{\ell=1}^{L} y^{(\ell)} P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta}) \quad (13.12)$$

$$\hat{\sigma}_k^2 = \frac{1}{s_k}\sum_{\ell=1}^{L} (y^{(\ell)} - \hat{m}_k)^2 P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta}) \quad (13.13)$$

$s_k \triangleq \sum_{\ell=1}^{L} P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta})$

(13.12) is a weighted avg. of samples $y^{(\ell)}$ weighted by $P(X^{(\ell)}=k \mid y^{(\ell)}, \hat{\theta})$

4. repeat 2.3. until convergence

13.1.3 Multivariate Case & Clustering

Generalization to Multivar.: replace (13.13) by an esti. of cov. mtx. with some regularization (3.25) $\hat{V}_Y = V_S + \frac{\rho}{L}\cdot I$ $\rho = f(L)$ e.g.

Then it can be viewed as improved k-means clustering: the "hard" assignments to classes is replaced by pos. (13.10) and the isotropic Euclidean distance by a general covariance matrix.

The also in 13.1.2 can be extended with a final clustering step that assigns ~~classes~~ samples to classes ("= distributions"?). Can handle many cases where k-means fails.

END