For today

- Linear State Space Models / Linear Dynamic Systems
- Kalman filter
- Smoothing
- Start with convex optimization intro.

Multivariate Gaussians refresher

$$\mathbf{z} = \left[\begin{array}{c} \mathbf{x} \\ \mathbf{y} \end{array}\right] \sim \mathcal{N}\left(\left[\begin{array}{c} \mathbf{a} \\ \mathbf{b} \end{array}\right], \left[\begin{array}{cc} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathbf{T}} & \mathbf{B} \end{array}\right]\right)$$

Marginals

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\mathbf{a}, \mathbf{A}) \\ \mathbf{y} &\sim \mathcal{N}(\mathbf{b}, \mathbf{B}) \end{aligned}$$

From a joint to conditionals

$$\begin{aligned} \mathbf{x}|\mathbf{y} &\sim \mathcal{N}\left(\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^{T}\right) \\ \mathbf{y}|\mathbf{x} &\sim \mathcal{N}\left(\mathbf{b} + \mathbf{C}^{T}\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^{T}\mathbf{A}^{-1}\mathbf{C}\right) \end{aligned}$$

$\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^{T}$ is Schur complement of the joint covariance matrix.

Multivariate Gaussians refresher

Forming a joint from marginal and conditional

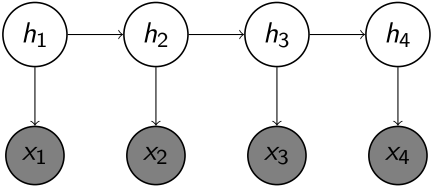
$$\begin{aligned}
\mathbf{x} &\sim \mathcal{N}(\mathbf{a}, \mathbf{A}) \\
\mathbf{y}|\mathbf{x} &\sim \mathcal{N}(\mathbf{b} + \mathbf{C}\mathbf{x}, \mathbf{B}) \\
\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} + \mathbf{C}\mathbf{a} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{A}^{\mathbf{T}}\mathbf{C}^{\mathbf{T}} \\ \mathbf{C}\mathbf{A} & \mathbf{B} + \mathbf{C}\mathbf{A}^{\mathbf{T}}\mathbf{C}^{\mathbf{T}} \end{bmatrix} \right)
\end{aligned}$$

Hidden Markov Model

We looked at the Hidden Markov Models specified by

- transition probabilities between hidden state variables $p(h_i|h_{i-1})$
- emission or observation probabilities $p(x_i|h_i)$

We assumed that the hidden variables h_i were discrete.[1]



[1] Note that we had an example of continuous $p(y_i|h_i)$ in one of our HMMs

State Space Models

Now we are going to lift the assumption on discreteness of the random variables $\mathbf{h}_i$.[2]

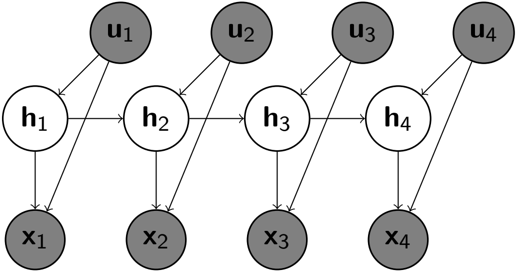
We still retain the conditional independence assumptions–the Bayes Net retains the same structure.

A particular choice of $p(\mathbf{h}_i|\mathbf{h}_{i-1})$ and $p(\mathbf{x}_i|\mathbf{h}_i)$ make the inference tractable, for example Gaussian densities.

[2] It is trivial to convert a multivariate discrete random variable into a univariate one–just list all the combinations–this is not the case with multivariate continuous variables.

Linear Dynamical System

We will also add a bit of flexibility to the model by introducing an input or control variable $\mathbf{u}$.



We will write the model in terms of structural equations

$$\begin{aligned}
\mathbf{h}_i &= \mathbf{A}_i\mathbf{h}_{i-1} + \mathbf{B}_i\mathbf{u}_i + \boldsymbol{\epsilon}_i \\
\mathbf{x}_i &= \mathbf{C}_i\mathbf{h}_i + \mathbf{D}_i\mathbf{u}_i + \boldsymbol{\nu}_i \\
\boldsymbol{\epsilon}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_i) \\
\boldsymbol{\nu}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i)
\end{aligned}$$

Linear Dynamical System

This is a very general model, since matrices $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{Q}_i, \mathbf{R}_i$ can, in principle, vary with i.

Some or all of these matrices may be specified ahead of time and we would not need to learn them.

If we do need to learn them we will need to constrain them in order to avoid overfitting.

A simple constraint is to remove dependence on i ($\mathbf{A}_i = \mathbf{A}, \forall i$).

Finally, even though this is a fairly general model, it is still jointly Gaussian in $\mathbf{h}, \mathbf{x}$

A simple LDS

We will assume a simple version of the LDS, one without the control signal $\mathbf{u}_i$.

Further we will consider the hidden state $\mathbf{h}$ to be composed of 2D positions and velocities.[3]

$$\mathbf{h}_i = \begin{bmatrix} h_{1,i} \\ h_{2,i} \\ v_{1,i} \\ v_{2,i} \end{bmatrix}$$

We can write down following equations

$$\begin{aligned} h_{1,i} &= h_{1,i-1} + v_{1,i-1}\Delta + \epsilon_{1,i} \\ h_{2,i} &= h_{2,i-1} + v_{2,i-1}\Delta + \epsilon_{2,i} \end{aligned}$$

[3] Video of bacteria on a plate, or mouse pointer on screen

Simple 2D LDS

In matrix form

$$\mathbf{h}_i = \begin{bmatrix} h_{1,i} \\ h_{2,i} \\ v_{1,i} \\ v_{2,i} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_i} \underbrace{\begin{bmatrix} h_{1,i-1} \\ h_{2,i-1} \\ v_{1,i-1} \\ v_{2,i-1} \end{bmatrix}}_{\mathbf{h}_{i-1}} + \boldsymbol{\epsilon}$$

This gives us underlying dynamics of the *hidden* state of the system. However, we observe **x** rather than **h** hence we need to specify how the measurements **x** depend on **h**.

$$\mathbf{x}_i = \mathbf{C}_i \mathbf{h}_i + \boldsymbol{\nu}_i$$

If we are tracking bacteria on a plate or a pointer on screen we only observe their location at different time points i, but not their velocities. Hence

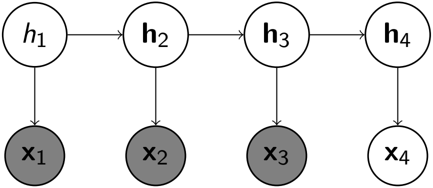
$$\mathbf{C}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Inference in LDS

Given measurements upto $\mathbf{x}_1, \ldots, \mathbf{x}_i$ we can ask some prototypical :

- Predict next hidden state $\mathbf{h}_{i+1}$, and consequently $\mathbf{x}_{i+1}$
- What is the distribution over the hidden states $\mathbf{h}_1, \ldots, \mathbf{h}_i$

Predicting next state



We have observations $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$ and we wish to predict $\mathbf{x}_i$.

Forward pass - Kalman filter

Just like with the HMM we will perform a forward pass.

First we note that $p(\mathbf{x}_1, \mathbf{h}_1, \ldots, \mathbf{x}_i, \mathbf{h}_i)$ is a Gaussian distribution, and so are its marginals, conditionals, and marginal conditionals.

Hence

$$p(\mathbf{h}_i | \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}) = \mathcal{N}(\boldsymbol{\mu}_{i|i-1}, \mathbf{S}_{i|i-1})$$

where we use notation $i|i-1$ to indicate that particular parameter is relevant to i^{th} slice and dependent on all observations up to $i-1$.

Forward pass - Kalman filter

Suppose we have computed the distribution of $\mathbf{h}_{i-1}$ given all the observations up to i-1

$$\mathbf{h}_{i-1}|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1} \sim \mathcal{N}(\boldsymbol{\mu}_{i-1|i-1}, \mathbf{S}_{i-1|i-1})$$

and recalling our model

$$\mathbf{h}_i = \mathbf{A}_i\mathbf{h}_{i-1} + \boldsymbol{\epsilon}_i$$

that is to say

$$\mathbf{h}_i|\mathbf{h}_{i-1} \sim \mathcal{N}(\mathbf{A}_i\mathbf{h}_{i-1}, \mathbf{Q}_i)$$

Now we can form a joint

$$\begin{bmatrix}\mathbf{h}_{i-1}\\ \mathbf{h}_i\end{bmatrix}\Bigg|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{i-1|i-1}\\ \mathbf{A}_i\boldsymbol{\mu}_{i-1|i-1}\end{bmatrix}, \begin{bmatrix}\mathbf{S}_{i-1|i-1} & \mathbf{S}_{i-1|i-1}^T\mathbf{A}_i^T\\ \mathbf{A}_i\mathbf{S}_{i-1|i-1} & \mathbf{A}_i\mathbf{S}_{i-1|i-1}\mathbf{A}_i^T + \mathbf{Q}_i\end{bmatrix}\right)$$

Forward pass - Kalman filter

From joint

$$\begin{bmatrix}\mathbf{h}_{i-1}\\ \mathbf{h}_i\end{bmatrix}\Bigg|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}\sim\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{i-1|i-1}\\ \mathbf{A}_i\boldsymbol{\mu}_{i-1|i-1}\end{bmatrix},\begin{bmatrix}\mathbf{S}_{i-1|i-1} & \mathbf{S}_{i-1|i-1}^T\mathbf{A}_i^T\\ \mathbf{A}_i\mathbf{S}_{i-1|i-1} & \mathbf{A}_i\mathbf{S}_{i-1|i-1}\mathbf{A}_i^T+\mathbf{Q}_i\end{bmatrix}\right)$$

we obtain marginal

$$\mathbf{h}_i|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}\sim\mathcal{N}(\boldsymbol{\mu}_{i|i-1},\mathbf{S}_{i|i-1})$$

where

$$\begin{aligned}\boldsymbol{\mu}_{i|i-1} &= \mathbf{A}\boldsymbol{\mu}_{i-1|i-1}\\ \mathbf{S}_{i|i-1} &= \mathbf{A}\mathbf{S}_{i-1|i-1}\mathbf{A}^T+\mathbf{Q}_i\end{aligned}$$

Forward pass - Kalman Filter

So we have

$$\mathbf{h}_i | \mathbf{x}_1, \ldots, \mathbf{x}_{i-1} \sim \mathcal{N}(\boldsymbol{\mu}_{i|i-1}, \mathbf{S}_{i|i-1})$$

and from our model

$$\mathbf{x}_i | \mathbf{h}_i \sim \mathcal{N}(\mathbf{C}_i \mathbf{h}_i, \mathbf{R}_i)$$

and now we can write a joint

$$\begin{bmatrix} \mathbf{h}_i \\ \mathbf{x}_i \end{bmatrix} \Bigg| \mathbf{x}_1, \ldots, \mathbf{x}_{i-1} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{i|i-1} \\ \mathbf{C}_i \boldsymbol{\mu}_{i|i-1} \end{bmatrix}, \begin{bmatrix} \mathbf{S}_{i|i-1} & \mathbf{S}_{i|i-1}^T \mathbf{C}_i^T \\ \mathbf{C}_i \mathbf{S}_{i|i-1} & \mathbf{C}_i \mathbf{S}_{i|i-1} \mathbf{C}_i^T + \mathbf{R}_i \end{bmatrix} \right)$$

Forward pass - Kalman Filter

$$\begin{bmatrix}\mathbf{h}_i\\ \mathbf{x}_i\end{bmatrix}\Bigg|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}\sim\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{i|i-1}\\ \mathbf{C}_i\boldsymbol{\mu}_{i|i-1}\end{bmatrix},\begin{bmatrix}\mathbf{S}_{i|i-1} & \mathbf{S}_{i|i-1}^T\mathbf{C}_i^T\\ \mathbf{C}_i\mathbf{S}_{i|i-1} & \mathbf{C}_i\mathbf{S}_{i|i-1}\mathbf{C}_i^T+\mathbf{R}_i\end{bmatrix}\right)$$

Now we can read marginal directly

$$\mathbf{x}_i|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}\sim\mathcal{N}(\mathbf{C}_i\boldsymbol{\mu}_{i|i-1},\mathbf{C}_i\mathbf{S}_{i|i-1}\mathbf{C}_i^T+\mathbf{R}_i)$$

And also we can obtain conditional

$$\mathbf{h}_i|\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\mathbf{x}_i\sim\mathcal{N}(\boldsymbol{\mu}_{i|i},\mathbf{S}_{i|i})$$

where

$$\begin{aligned}\boldsymbol{\mu}_{i|i} &= \boldsymbol{\mu}_{i|i-1}+\mathbf{S}_{i|i-1}\mathbf{C}_i^T(\mathbf{C}_i\mathbf{S}_{i|i-1}\mathbf{C}_i^T+\mathbf{R})^{-1}(\mathbf{x}_i-\mathbf{C}_i\boldsymbol{\mu}_{i|i-1})\\ \mathbf{S}_{i|i} &= \mathbf{S}_{i|i-1}-\mathbf{S}_{i|i-1}\mathbf{C}_i^T(\mathbf{C}_i\mathbf{S}_{i|i-1}\mathbf{C}_i^T+\mathbf{R})^{-1}\mathbf{C}_i\mathbf{S}_{i|i-1}\end{aligned}$$

Forward pass - Kalman Filter

Now we can write out all of the recursive rules

$$\begin{aligned}
\boldsymbol{\mu}_{i|i-1} &= \mathbf{A}_i \boldsymbol{\mu}_{i-1|i-1} \\
\mathbf{S}_{i|i-1} &= \mathbf{A}_i \mathbf{S}_{i-1|i-1} \mathbf{A}_i^T \boldsymbol{\mu}_{i-1|i-1} \\
\mathbf{K}_i &= \mathbf{S}_{i|i-1} \mathbf{C}_i^T (\mathbf{C}_i \mathbf{S}_{i|i-1} \mathbf{C}_i^T + R)^{-1} \\
\boldsymbol{\mu}_{i|i} &= \boldsymbol{\mu}_{i|i-1} + \mathbf{K}_i (\mathbf{x}_i - \mathbf{C}_i \boldsymbol{\mu}_{i|i-1}) \\
\mathbf{S}_{i|i} &= \mathbf{S}_{i|i-1} - \mathbf{K}_i \mathbf{C}_i \mathbf{S}_{i|i-1}
\end{aligned}$$

with the initial $\boldsymbol{\mu}_{1|0} = \mathbf{0}$ and $\mathbf{S}_{i,i} = c\mathbf{I}$ for a large c.

Backward pass - Smoothing

Note that the updates we derived were all conditioned on the data up to and including i slice.

The mean $\boldsymbol{\mu}_{i|i}$ and covariance $\mathbf{S}_{i|i}$ do not depend on the data after i^{th} slice. Recall we only computed $p(\mathbf{h}_i|\mathbf{x}_1, \dots, \mathbf{x}_i)$.

In case of HMM, we had both forward and backward pass and we computed

$$p(h_i|x_1, \dots, x_i, \dots, x_n) \propto \alpha(h_i)\beta(h_i)$$

We lack a backward pass for the LDS model.

Backward pass - Smoothing

There are different smoothing algorithms we will use RTS (Rauch-Tung-Striebel).

We will skip the derivation here and just give the backward pass updates

$$\begin{aligned}
\mathbf{L}_i &= \mathbf{S}_{i|i}\mathbf{A}_i^T\mathbf{S}_{i+1|i}^T \\
\boldsymbol{\mu}_{i|N} &= \boldsymbol{\mu}_{i|i} + \mathbf{L}_i(\boldsymbol{\mu}_{i+1|N} - \boldsymbol{\mu}_{i+1|i}) \\
\mathbf{S}_{i|N} &= \mathbf{S}_{i|i} + \mathbf{L}_i(\mathbf{S}_{i+1|N} - \mathbf{S}_{i+1|i})\mathbf{L}_i^T
\end{aligned}$$

We note that unlike in the case of HMM, the backward pass depends on the output of forward pass.

The initialization of $\boldsymbol{\mu}_{N|N}, \mathbf{S}_{N|N}$ are results of the forward pass.

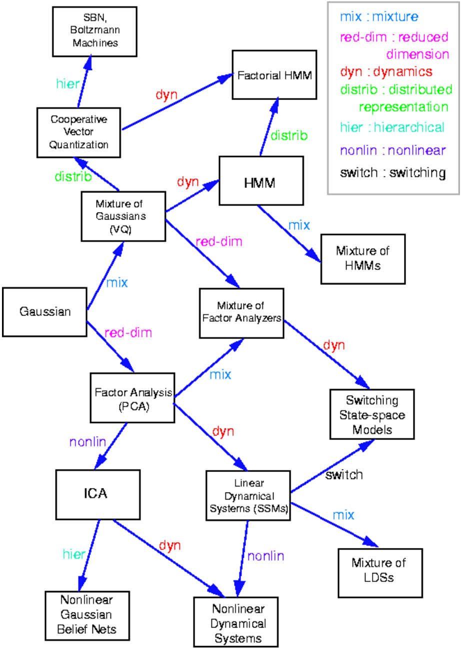
Most likely assignment

Once we have completed the backward pass we can answer that depend on all of the observations.

For example, the most likely hidden state at time i given all the data is given by $\boldsymbol{\mu}_{i|N}$.

The uncertainty around this state is described by the covariance $\mathbf{S}_{i|N}$

Roweis S, Ghahramani Z, A Unifying Review of Linear Gaussian Models



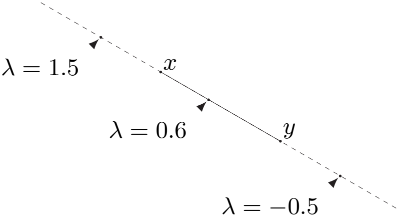
Fast intro to convex optimization

We will now go through some basic concepts in convex optimization.

Affine sets

A set $S \subseteq \mathbf{R}^n$ is **affine** if

$$x, y \in S, \lambda, \mu \in \mathbf{R}, \lambda + \mu = 1 \Rightarrow \lambda x + \mu y \in S$$



Other ways of specifying affine sets

$$\begin{aligned} S &= \{Az + b | z \in \mathbf{R}^q\} \\ S &= \{x | Bx = d\} \end{aligned}$$

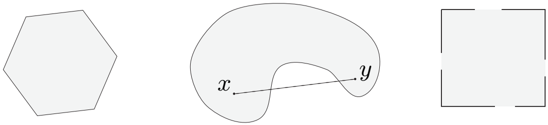
Convex sets

A set $S \subseteq \mathbf{R}^n$ is **convex** if

$$x, y \in S, \lambda, \mu \geq 0, \lambda + \mu = 1 \Rightarrow \lambda x + \mu y \in S$$

convex

not convex

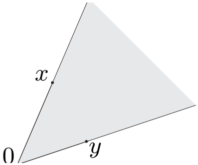


Convex sets are affine because $\mathbf{R}_+ = \{x | x \geq 0\} \subset \mathbf{R}$.

Convex sets - cones

A set $S \subseteq \mathbf{R}^n$ is a **convex cone** if

$$x, y \in S, \lambda, \mu \geq 0 \Rightarrow \lambda x + \mu y \in S$$



Nonnegative orthant $\mathbf{R}^n_+$ is a cone.

Set of positive semidefinite matrices is a cone

$$\mathbf{S}^n_+ = \{X \in \mathbf{S}^n | \forall a \in \mathbf{R}^n, a^T X a \geq 0\}$$

where $\mathbf{S}^n$ is the set of symmetric matrices of size $n \times n$.

$$a^T(\lambda X + \mu Y)a = \lambda a^T X a + \mu a^T Y a \geq 0$$

Combinations

For $x_1, \dots x_k \in \mathbf{R}^n$ and $\theta \in \mathbf{R}^k$ and $y = \sum_{i=1}^{k} \theta_i x_i$

y is a	$\sum_i \theta_i = 1$	$\theta_1, \dots \theta_k \geq 0$
Linear combination		
Affine combination	✓	
Conic combination		✓
Convex combination	✓	✓

Hyperplanes and halfspaces

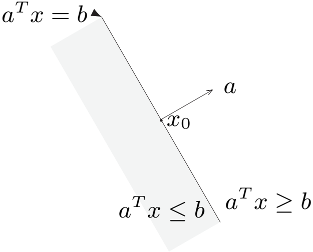
A **hyperplane** is defined as

$$\{x|a^T x = b\}$$

and we assume $a \neq \mathbf{0}$.
A **halfspace** is defined as

$$\{x|a^T x \leq b\}$$

and we assume $a \neq \mathbf{0}$.



Closure under intersection

$$\forall i \in I, S_i \text{ is } \begin{pmatrix} \text{affine} \\ \text{convex} \\ \text{convex cone} \end{pmatrix} \Longrightarrow \bigcap_{i \in I} S_i \begin{pmatrix} \text{affine} \\ \text{convex} \\ \text{convex cone} \end{pmatrix}$$

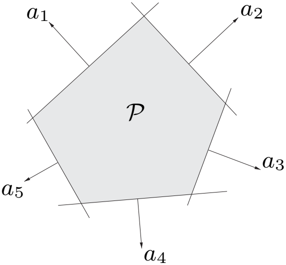
Convex sets as halfspace intersections

In general a convex set S

$$S = \bigcap \{\mathcal{H} | \mathcal{H} \text{ is halfspace}, S \subseteq \mathcal{H}\}$$

A simple example **polyhedron**

$$\mathcal{P} = \left\{ x | a_i^T x \leq b_i, i = 1, \ldots, k \right\}$$



Boundary of a set

We need a concept of an ϵ-ball

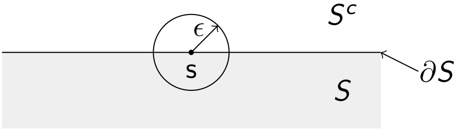
$$B(s, \epsilon) = \{x | \, \|x - s\| \leq \epsilon\}$$

all points within ϵ from s.

For a set $S \subseteq \mathbf{R}^n$ we say that $s \in S$ is a **boundary point** ($s \in \partial S$) if for all $\epsilon \geq 0$

$$B(s, \epsilon) \cap S^c \neq \emptyset$$

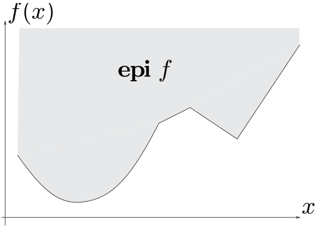
where $S^c := \{x \in \mathbf{R}^n, x \notin S\}$



Convex functions – epigraph

Given a function $f : \mathbf{R}^n \to \mathbf{R}$ the **epigraph** of a function is a subset of $\mathbf{R}^n \times \mathbf{R} = \mathbf{R}^{n+1}$

$$\mathbf{epi}(f) = \{(x, y) | y \geq f(x), x \in \mathbf{R}^n\}$$



A function $f : \mathbf{R}^n \to \mathbf{R}$ is a **convex function** if $\mathbf{epi}(f)$ is a convex set.

Convex functions – domain

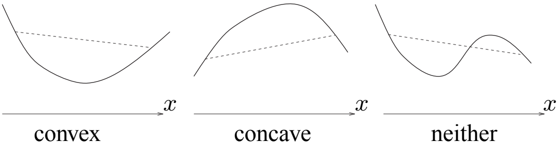
For a function $f : \mathbf{R}^n \to \mathbf{R}$ we will use $\mathbf{dom}(f) \subseteq \mathbf{R}^n$ to denote the set on which function f is defined.

Convex functions

A function $f : \mathbf{R}^n \to \mathbf{R}$ is a **convex function** if for all $x, y \in \mathbf{dom}(f)$ and $\lambda \in [0, 1]$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

A function $f : \mathbf{R}^n \to \mathbf{R}$ is a **concave function** if $-f$ is a convex function.



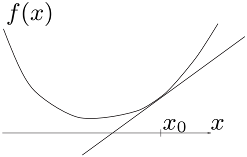
Convex function – first order characterization

Recall Taylor expansion

$$f(x) \approx f(x_0) + (x - x_0)^T \nabla f(x_0)$$

A differentiable function $f : \mathbf{R}^n \to \mathbf{R}$ is convex if and only if for all $x, x_0 \in \mathbf{dom}(f)$

$$f(x) \geq f(x_0) + (x - x_0)^T \nabla f(x_0)$$



Convex function – second order characterization

Recall Taylor expansion

$$f(x) \approx f(x_0) + (x - x_0)^T \nabla f(x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0)$$

A twice-differentiable function $f : \mathbf{R}^n \to \mathbf{R}$ is convex if and only for all $x \in \mathbf{dom}(f)$, $\nabla^2 f(x) \in \mathbf{S}_+^n$ (Hessian is positive semi-definite)

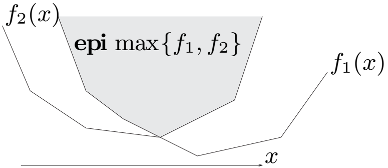
Convex function examples

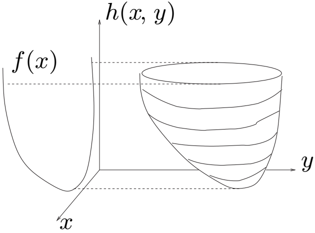
- x^α for $x \in \mathbf{R}_+, \alpha \geq 1$
- $x \log x$ for $x \in \mathbf{R}_+$
- $\log \sum_i \exp x_i$
- $a^T x + b$ for $a \in \mathbf{R}^n, b \in \mathbf{R}$
- $x^T P x + 2q^T x + r$ for $P \in \mathbf{S}_+$

Operations that retain convexity

If f, g, h_i are convex then so is

- $\alpha_1 f + \alpha_2 g$ for $\alpha_1, \alpha_2 \geq 0$
- $\int_y p(y) f(x, y) dy$ if $\forall y, p(y) \geq 0$
- $\max_i h_i$
- $\inf_y f(x, y)$
- $f(Ax + b)$





Convex optimization – standard form

The problem

$$
\begin{aligned}
\underset{x\in\mathcal{D}}{\text{minimize}}\ & f_0(x) \\
\text{subject to}\ & f_i(x) \leq 0, i = 1, \ldots, m \\
& h_i(x) = 0, i = 1, \ldots, p
\end{aligned}
$$

where $\mathcal{D} = \bigcap_{i=0}^{p} \mathbf{dom}(f_i) \cap \bigcap_{i=1}^{m} \mathbf{dom}(h_i)$.
If f_i are all convex and h_i are all affine then the above problem is convex.

For today

- Linear State Space Models / Linear Dynamic Systems
- Kalman filter
- Smoothing
- Started with convex optimization intro.