Seminar talk series:
Machine Learning for humen-computer interaction

Gaussian Processes for Machine Learning

An introduction to Gaussian Processes, (scaled) GPLVMs, (balanced) GPDMs and their applications to 3D people tracking

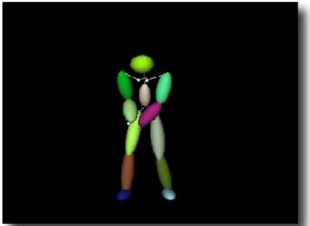
Andreas Geiger

Interactive System Laboratories (interACT)
Fakultät für Informatik, Universität Karlsruhe (TH)

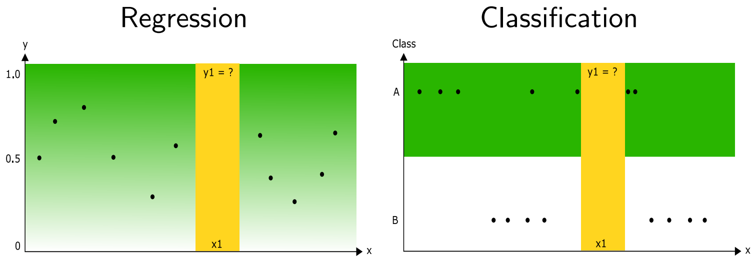
June 20, 2007

Applications of Gaussian Process theory

- CO_2 concentration forecast
- Handwriting recognition
- Determining trustworthiness of bank clients
- Focussing multiple-mirror telescopes
- Generating music playlists
- Articulated body tracking:

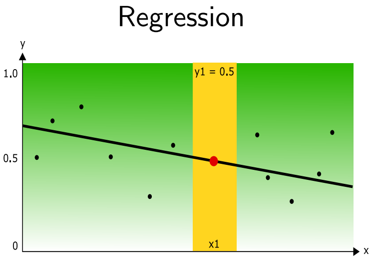


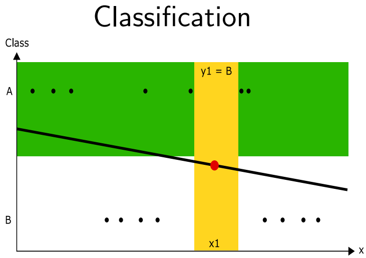
Problem: How to fit a line or curve to some given data?



- **Input:** Training data $\{(x_n, y_n)\}_{n=1}^{N}$ and Query $\{x_m\}_{m=1}^{M}$
- **Output:** Prediction $\{y_m\}_{m=1}^{M}$
- x represents source data and y represents target data
- Regression: $y \in \mathbb{R}$, Classification: $y \in Class$ $(|Class| < \infty)$

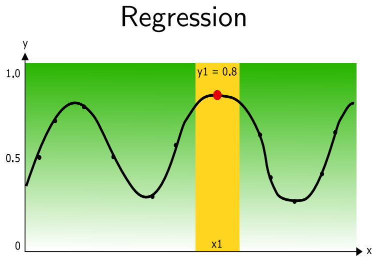
Linear regression and classification





- Chose y as a **linear function** of x
- Example: $y = ax + b$
- Task: Determine parameters a and b
- Not suitable in this case $\rightarrow$ We need something more general!

Non-linear regression and classification

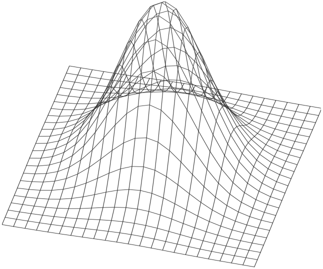


- Chose y as a **non-linear function** of x
- Example: $y = w_0 x^0 + w_1 x^1 + \ldots + w_n x^n$
- Task: Determine parameters w_i
- More suitable, but difficult to determine paramters!

How to solve this problem?

- Parametric approaches
  - Polynomials
  - Piecewise polynomials (Splines)
  - Neural Networks
  - Support Vector Machines
- Non-parametric approaches
  - K-nearest neighbors
  - Gaussian Processes

Can all this be done by a simple gaussian distribution?



What is a Gaussian Process?

What is a Gaussian Process?

Does it address the production of german 10-Mark notes?

No, probably not ;)

Definition

Gaussian Process: A Collection of normally distributed random variables

A Gaussian process is a stochastic process which generates samples over time $\{X_t\}_{t \in T}$ such that no matter which finite linear combination of X_t ones takes, that linear combination will be normally distributed.

Stochastic Process: A Collection of random variables

Let $(\Omega, \mathcal{F}, P)$ be a probability space, $(Z, \mathcal{Z})$ a space with σ-algebra and T a set of indices. A stochastic process X is defined as

$$X : \Omega \times T \rightarrow Z, \; (\omega, t) \mapsto X_t(\omega)$$

with random variables $X_t : \Omega \rightarrow Z$ for all $t \in T$.

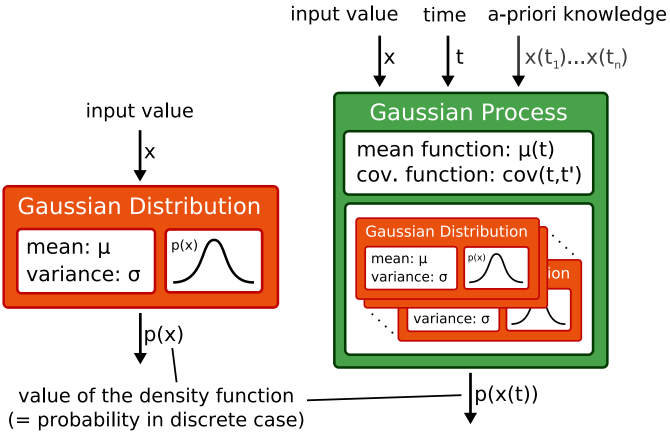
Definition

Probability Space: Samples, Events, Probability measure

A probability space $(\Omega, \mathcal{F}, P)$ is a measure space with a measure P that satisfies the probability axioms.

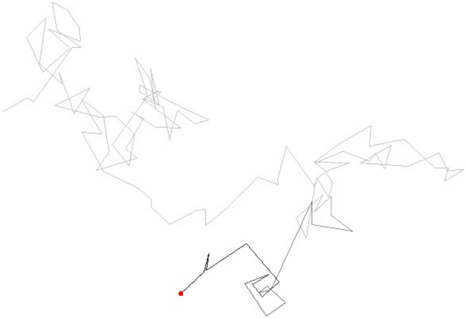
- The sample space Ω, is a nonempty set of samples ω.
- The event space $\mathcal{F}$ is a σ-algebra of subsets of Ω. Its elements are called events, which are sets of outcomes for which one can ask a probability.
- The probability measure P is a function from $\mathcal{F}$ to the real numbers.

Gaussian distribution vs. Gaussian Process



$$X \sim \mathcal{N}(\mu, \sigma) \qquad X(t) \sim \mathcal{GP}(\mu(t), cov(t, t'))$$

A Gaussian Process example: Brownian Motion



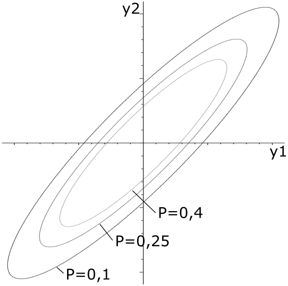
The Brownian Motion (= Wiener Process) is a Gaussian Process

$X(t) - X(t') \sim \mathcal{N}(0, t - t')$ $\qquad$ $X(t) \sim \mathcal{GP}(0, min(t, t'))$

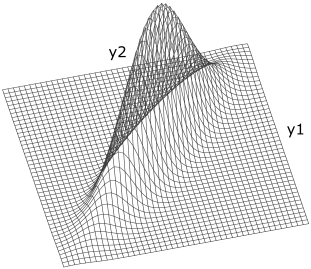
Joint distribution of strongly correlated $y = (y1, y2)$

Zero-mean 2D gaussian distribution

Contour plot



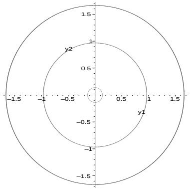
3D plot



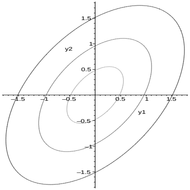
$$P(y|K) = \frac{1}{\sqrt{2\pi|K|}} e^{-\frac{1}{2}y^T K^{-1} y} \qquad K = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

Influence of the covariance matrix entries

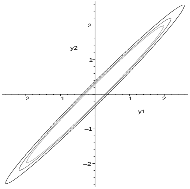
These are some contour plots of 2D gaussian distributions with different covariance matrices. Covariance is a measure of how much two random variables vary together. 1 means perfect linear coherence, -1 means perfect negative linear coherence. If it is 0 there is no linear coherence.



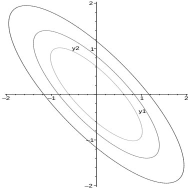
$$K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$K = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

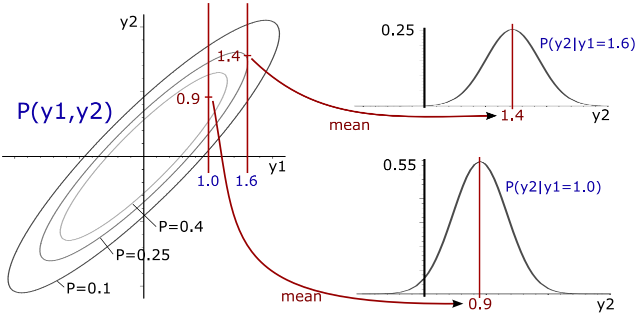


$$K = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}$$



$$K = \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}$$

Conditional distribution $P(y2|y1)$



Let us assume that we know the covariance matrix K and y_1. The posteriori distribution $P(y2|y1)$ is a gaussian, too. Our job is now to determine the mean $\overline{y_2}$ and the corresponding variance $\widetilde{y_2}$!

Determining the mean $\overline{y_2}$ and the variance $\widetilde{y_2}$

$$
\begin{aligned}
P(y_2|y_1,K) &= \frac{P(y_1,y_2|K)}{P(y_1|K)} && (1)\\
&\propto exp - \frac{1}{2}\left\{\begin{pmatrix} y_1 & y_2 \end{pmatrix}\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right\} && (2)\\
&= exp - \frac{1}{2}\left\{y_1^2 a + 2y_1 y_2 b + y_2^2 c\right\} && (3)\\
&\propto exp - \frac{1}{2}\left\{2y_1 y_2 b + y_2^2 c\right\} && (4)\\
&= exp - \frac{1}{2}\left\{(y_2^2 + 2y_2 y_1 \frac{b}{c})c\right\} && (5)\\
&\propto exp - \frac{1}{2}\left\{(y_2^2 + 2y_2 y_1 \frac{b}{c} + y_1^2 \frac{b^2}{c^2})c\right\} && (6)\\
&= exp - \frac{1}{2}\left\{((y_2 + y_1 \frac{b}{c})^2)c\right\} && (7)\\
&= exp - \frac{1}{2}\left\{\frac{(y_2 - (-y_1 \frac{b}{c}))^2}{1/c}\right\} && (8)
\end{aligned}
$$

Determining the mean $\overline{y_2}$ and the variance $\widetilde{y_2}$

$\Rightarrow$ **Mean** $\overline{y_2} = -y_1 \frac{b}{c}$**, variance** $\widetilde{y_2} = \frac{1}{c}$

Annotations on the slide before:

We assume the inverse of the covariance matrix $K^{-1} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$

(1) $\rightarrow$ (2): Since we've selected y_1 fix we know that $P(y_1|K)$ is a constant. We're interested only in the distribution (with $\int P(y_2|y_1, K)dy_2 = 1$), so this constant can be neglect.

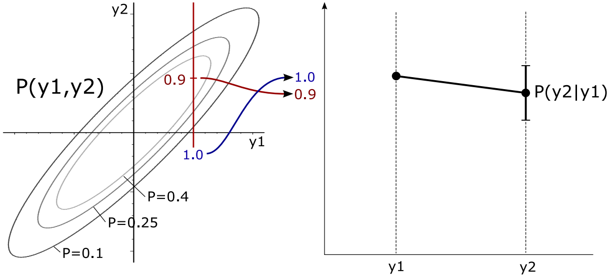
(3) $\rightarrow$ (4): $y_1^2 a$ can be factored out since it is an additive component of the exponent. It is a constant so may also neglect it.

(5) $\rightarrow$ (6): We expand the term by an additive constant $y_1^2 \frac{b^2}{c^2}$ which is allowed for the reasons above.

A new representation for our example

Let K be $\begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$ and assume $y_1 = 1.0$

Then we get $K^{-1} = \begin{pmatrix} 5.26 & -4.74 \\ -4.74 & 5.26 \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$. Now we are able to calculate the mean and variance of $P(y_2|y_1, K)$, following the equations above: $\overline{y_2} = -y_1 \frac{b}{c} = 0.9 \qquad \widetilde{y_2} = \frac{1}{c} = 0.19$



Extending our approach to vectors ($\overrightarrow{y_1}$ and $\overrightarrow{y_2}$)

Up to now we've found a representation for our 2 scalars y_1 and y_2 where y_1 was the known input data and y_2 was the requested output data which we represented by its mean and variance!

How can we extend this approach to cover multi-dimensional vectors $\overrightarrow{y_1}$ and $\overrightarrow{y_2}$?

Therefore let us now assume K^{-1} as $\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{m \times m}$. Let further be $y_1 \in \mathbb{R}^n$ and $y_2 \in \mathbb{R}^m$.

By generalizing the equations above we get ...

Extending our approach to vectors ($\overrightarrow{y_1}$ and $\overrightarrow{y_2}$)

$$
\begin{aligned}
P(y_2|y_1, K) &= \frac{P(y_1, y_2|K)}{P(y_1|K)} && (9)\\
&\propto exp - \frac{1}{2}\left\{\begin{pmatrix} y_1^T & y_2^T \end{pmatrix}\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right\} && (10)\\
&= exp - \frac{1}{2}\left\{y_1^T A y_1 + y_2^T B^T y_1 + y_1 T B y_2 + y_2^T C y_2\right\} && (11)\\
&\propto exp - \frac{1}{2}\left\{y_2^T C y_2 + y_2^T B^T y_1 + y_1^T B y_2\right\} && (12)\\
&\propto exp - \frac{1}{2}\left\{y_2^T C y_2 + y_2^T B^T y_1 + y_1^T B y_2 + y_1^T B C^{-1} B^T y_1\right\} && (13)\\
&= exp - \frac{1}{2}\left\{(y_2^T C + y_1^T B)(y_2 + C^{-1} B^T y_1)\right\} && (14)\\
&= exp - \frac{1}{2}\left\{(y_2^T + y_1^T B C^{-1}) C (y_2 + C^{-1} B^T y_1)\right\} && (15)\\
&= exp - \frac{1}{2}\left\{(y_2 - (-C^{-1} B^T y_1)) C (y_2 - (-C^{-1} B^T y_1))\right\} && (16)
\end{aligned}
$$

Extending our approach to vectors ($\overrightarrow{y_1}$ and $\overrightarrow{y_2}$)

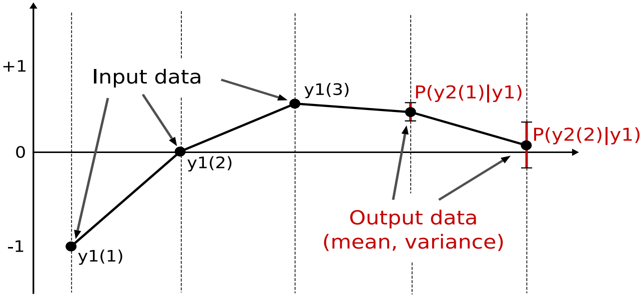
$$\begin{aligned} &= exp - \frac{1}{2}\left\{(y_2 - (-C^{-1}B^T y_1))C(y_2 - (-C^{-1}B^T y_1))\right\} \\ &= exp - \frac{1}{2}\left\{(y_2 - \overline{y_2})\widetilde{Y_2}(y_2 - \overline{y_2})\right\} \end{aligned} \tag{17}$$

with **mean** $\overline{y_2} = -C^{-1}B^T y_1$ and **variance** $\widetilde{Y_2} = C^{-1}$.

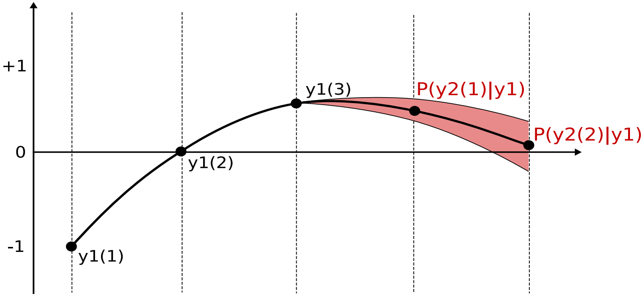
Extending our approach to vectors ($\overrightarrow{y_1}$ and $\overrightarrow{y_2}$)

Let now K be $\begin{pmatrix} 1.0 & 0.9 & 0.7 & 0.4 & 0.2 \\ 0.9 & 1.0 & 0.9 & 0.7 & 0.4 \\ 0.7 & 0.9 & 1.0 & 0.9 & 0.7 \\ 0.4 & 0.7 & 0.9 & 1.0 & 0.9 \\ 0.2 & 0.4 & 0.7 & 0.9 & 1.0 \end{pmatrix}$ and assume $y_1 = \begin{pmatrix} -1.0 \\ 0 \\ 0.5 \end{pmatrix}$.

Using the equations above we get $\overline{y_2} = \begin{pmatrix} 0.43 & 0.1 \end{pmatrix}^T$ and $\widetilde{Y_2} = \begin{pmatrix} 0.04 & 0.09 \\ 0.09 & 0.24 \end{pmatrix}$:



Extending our approach to vectors ($\overrightarrow{y_1}$ and $\overrightarrow{y_2}$)



Now, doesn't this look like non-linear regression?
But where did the 5x5 covariance matrix come from?

How the covariance matrix was made

How to build an appropriate covariance matrix?

Assumptions

We assume that points which are lying close together are strongly correlated. So we assign them a covariance close to 1. Points far from each other are only weakly correlated. Thus we assign them a covariance close to 0.

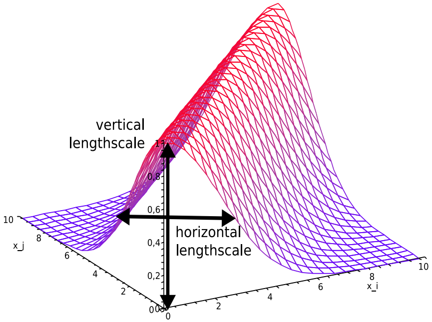
Such a covariance function can be defined by using a RBF:

$$Cov(y_i, y_j) = \sigma_f^2 e^{-\frac{1}{2l^2}(x_i - x_j)^2} + \sigma_\nu^2 \delta_{ij}$$

- σ_ν^2: noise
- l: horizontal lengthscale
- σ_f^2: vertical lengthscale

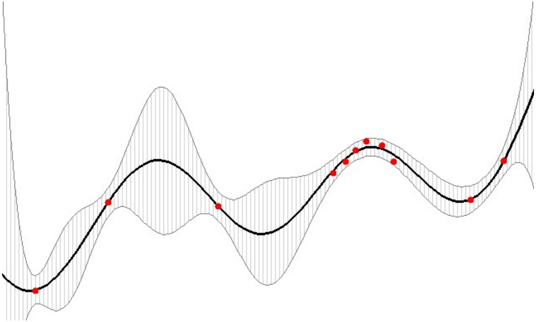
The Radial Basis Function (RBF) kernel

$$Cov(y_i, y_j) = \sigma_f^2 e^{-\frac{1}{2l^2}(x_i - x_j)^2} + \sigma_\nu^2 \delta_{ij}$$



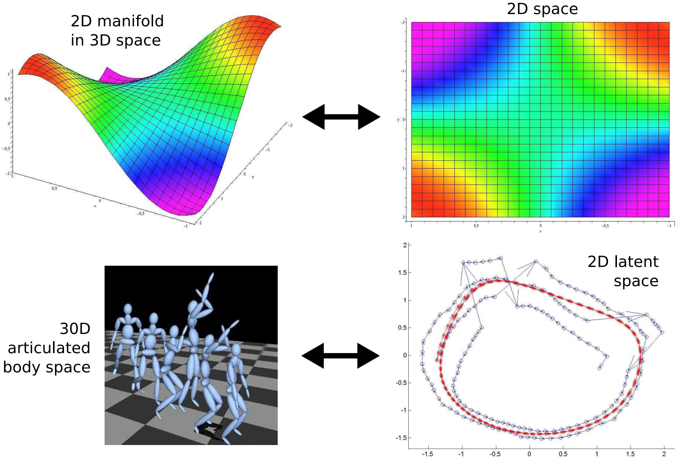
The hyperparameters σ_ν^2, l and σ_f^2 can be set manually or they can be found by maximizing the marginal likelihood $p(y|x, \sigma_\nu^2, l, \sigma_f^2)$.

A GP regression example



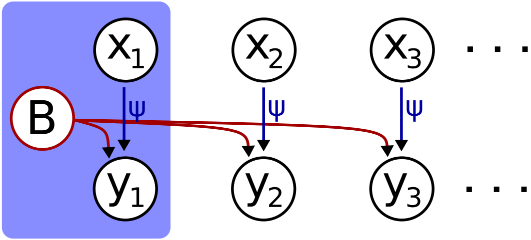
A regression curve plot by the "Gaussian Process Regression Applet" using 11 data points. One can observe that uncertainty goes down when multiple data points are aggregated together.

Our goal: Non-linear dimensionality reduction



A graphical formulation of dimensionality reduction

At each time step t we express our observations y as a combination of basis functions ψ of latent variables x.



$$y_t = \sum_j b_j \psi_j(x_t) + \delta_t \qquad (e.g.\ \psi_j(x) = x)$$

A formulation of linear dimensionality reduction

Let $\tilde{Y} = [\tilde{y}_1 ... \tilde{y}_N]$ be a set of D-dimensional data variables and let $\tilde{X} = [\tilde{x}_1 ... \tilde{x}_N]$ be a set of L-dimensional latent variables.

We now formulate a **mapping from latent to data space** by

$$\tilde{Y} = \tilde{B}\tilde{X} + \tilde{\Delta} \quad (\tilde{y}_n = \tilde{B}\tilde{x}_n + \tilde{\delta}_n)$$

where B^T is a design matrix (representing the linear mapping) and Δ^T the noise term. The dual problem is

$$Y = XB + \Delta \quad (y_d = Xb_d + \delta_d)$$

where $X^T = \tilde{X}$ and x_d represents the dth column of X.

Marginalizing over the parameters B

We now marginalize over the parameters B:

$$P(Y|X,\Delta) = \prod_{d=1}^{D} p(y_d|X,\delta_d) \tag{18}$$

$$= \prod_{d=1}^{D} \int_{\mathbb{R}^L} p(y_d|X,b_d,\delta_d)p(b_d)db_d \tag{19}$$

Bayesian methodology requires us to select suitable priors:

$$b_d \sim \mathcal{N}(0,I) \qquad \delta_d \sim \mathcal{N}(0,\beta^{-1}I)$$

Calculating the mean and variance of $y_d|X, \delta_d$

Marginalizing with Gaussian priors yields a Gaussian distribution. We only need to calculate the mean and variance of $y_d|X, \delta_d$.

$$
\begin{aligned}
Mean(y_d) &= \overline{y_d} = \mathcal{E}\left\{Xb_d + \delta_d\right\} \\
&= X\mathcal{E}\left\{b_d\right\} + \mathcal{E}\left\{\delta_d\right\} = 0 \\
Cov(y_d) &= \mathcal{E}\left\{(y_d - \overline{y_d})(y_d - \overline{y_d})^T\right\} = \mathcal{E}\left\{y_d y_d^T\right\} \\
&= \mathcal{E}\left\{(Xb_d + \delta_d)(Xb_d + \delta_d)^T\right\} \\
&= X\mathcal{E}\left\{b_d b_d^T\right\} X^T + \mathcal{E}\left\{\delta_d \delta_d^T\right\} = XX^T + \beta^{-1} I
\end{aligned}
$$

$$\Rightarrow y_d|X, \delta_d \sim \mathcal{N}(0, XX^T + \beta^{-1} I)$$

Maximizing the log-likelihood $\mathcal{L}$

With this result we can calculate the log-likelihood

$$\mathcal{L} = log\ p(Y|X,\Delta) = log \prod_{d=1}^{D} p(y_d|X,\delta_d) = \sum_{d=1}^{D} log\ p(y_d|X,\delta_d)$$
$$= const - \frac{D}{2} log|K| - \frac{1}{2} tr(K^{-1}YY^T)$$

where $K = XX^T + \beta^{-1}I$. It can be shown that this likelihood is maximized by $X = UZV^T$ with U containing the first L eigenvectors, Z is a $L \times L$ diagonal matrix with $z_{ll} = (\lambda_l - \frac{1}{\beta})^{-\frac{1}{2}}$ and V being an arbitrary $L \times L$ rotation matrix. With a richer non-linear kernel K gradient based optimization has to be used.

Scaled Gaussian Process Latent Variable Models

Accounting for the different scales in each dimension a weight matrix $W = diag(w_1, ..., w_D)$ is introduced. This yields

$$p(Y|M) = \frac{|W|^N}{\sqrt{(2\pi)^{ND}|K|^D}} exp(-\frac{1}{2} tr(K^{-1} Y W^2 Y^T))$$

where $M = \left\{ \{x_n\}_{n=1}^{N}, \{\beta_i\}_{i=1}^{3}, \{w_d\}_{d=1}^{D} \right\}$ are model parameters and a Radial Basis Function (RBF) is used as kernel:

$$k(x_i, x_j) = \beta_1 exp(-\frac{\beta_2}{2} \|x_i - x_j\|^2) + \frac{\delta(x_i, x_j)}{\beta_3}$$

Putting it all together: Maximizing the posterior

The posterior density over the model M is:

$$p(M|Y) \propto p(Y|M)p(M) = p(Y|M)p(X)p(\beta)p(W)$$

By specifying the priors

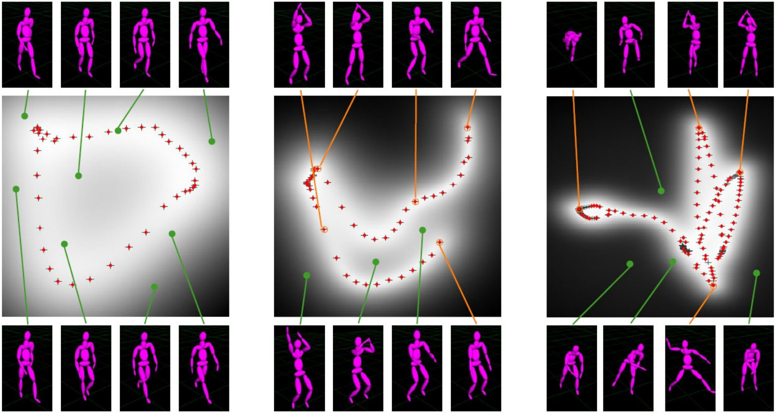
$$p(X) = \prod_{n=1}^{N} \mathcal{N}(x_n|0, I) \quad p(\beta) \propto \prod_{i=1}^{3} \frac{1}{\beta_i} \quad p(W) \propto 1$$

the log posterior gets $(k(x_i, x_j) = \beta_1 exp(-\frac{\beta_2}{2}\|x_i - x_j\|^2) + \frac{\delta(x_i,x_j)}{\beta_3})$

$$\mathcal{L} = -\frac{D}{2} log|K| - \frac{1}{2} tr(K^{-1}YW^2Y^T) - \frac{1}{2}\sum_{n=1}^{N}\|x_n\|^2 - \sum_{i=1}^{3} log(\beta_i) + Nlog|W|$$

which we maximize to learn the model $M = \{X, \beta\}$.

An example: Style-based inverse kinematic (Grochow)

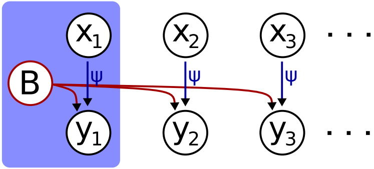


Learned GPLVMs using a Walk, a jump shot and a baseball pitch!

GPLVMs vs. GPDMs

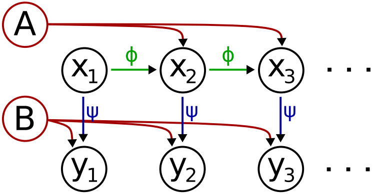
When switching from GPLVM to GPDM we take the dynamics (expressed by time t) into account:

GPLVM



$$y_t = \sum_j b_j \psi_j(x_t) + \delta_t$$

GPDM



$$x_t = \sum_i a_i \phi_j(x_{t-1}) + \delta_{x,t}$$
$$y_t = \sum_j b_j \psi_j(x_t) + \delta_{y,t}$$

Modeling the GPDM mapping and dynamics

The GPDM **dynamics prior** and kernel ($X_{out} = (x_2, ..., x_N)^T$):

$$p(X|\alpha) = \frac{p(x_1)}{\sqrt{(2\pi)^{(N-1)L}|K_X|^L}} exp(-\frac{1}{2} tr(K_X^{-1} X_{out} X_{out}^T))$$

$$k(x_i, x_j) = \alpha_1 exp(-\frac{\alpha_2}{2} \|x_i - x_j\|^2) + \alpha_3 x_i^T x_j + \frac{\delta(x_i, x_j)}{\alpha_4}$$

The GPDM **mapping prior** and kernel (same as in SGPLVM):

$$p(Y|X, \beta, W) = \frac{|W|^N}{\sqrt{(2\pi)^{ND}|K_Y|^D}} exp(-\frac{1}{2} tr(K_Y^{-1} Y W^2 Y^T))$$

$$k(x_i, x_j) = \beta_1 exp(-\frac{\beta_2}{2} \|x_i - x_j\|^2) + \frac{\delta(x_i, x_j)}{\beta_3}$$

Putting it all together: Maximizing the posterior

The posterior density over this new model is:

$$p(X, \alpha, \beta, W|Y) \propto p(Y|X, \beta, W)p(X|\alpha)p(\alpha)p(\beta)p(W)$$

By specifying the priors

$$p(\alpha) \propto \prod_{i=1}^{4} \frac{1}{\alpha_i} \qquad p(\beta) \propto \prod_{i=1}^{3} \frac{1}{\beta_i} \qquad p(W) \propto 1$$

we get the log posterior (up to an additive constant):

$$\begin{aligned} \mathcal{L} \quad = \quad & -\frac{L}{2} log|K_X| - \frac{1}{2} tr(K_X^{-1} X_{out} X_{out}^T) \\ & -\frac{D}{2} log|K_Y| - \frac{1}{2} tr(K_X^{-1} Y W^2 Y^T) + N log|W| \\ & -\sum_{i=1}^{4} log(\alpha_i) - \sum_{i=1}^{3} log(\beta_i) \end{aligned}$$

Balanced Gaussian Process Dynamical Models (B-GPDMs)

The B-GPDM introduces a factor $\lambda = \frac{D}{L}$ to balance the influence of the dynamics and the pose reconstruction by raising the dynamics density function to the ratio of their dimensions.

$$
\begin{aligned}
-\mathcal{L} \quad = \quad & \frac{D}{L}\left(\frac{L}{2} log|K_X| + \frac{1}{2} tr(K_X^{-1} X_{out} X_{out}^T)\right) \\
& + \frac{D}{2} log|K_Y| + \frac{1}{2} tr(K_Y^{-1} Y W^2 Y^T) - N log|W| \\
& + \sum_{i=1}^{4} log(\alpha_i) + \sum_{i=1}^{3} log(\beta_i)
\end{aligned}
$$

The Model $M = \{X, Y, \alpha, \beta, W\}$ is learned by minimizing $-\mathcal{L}$!

A tracking formulation

Given a model M and an image sequence $I_{1:T}$ we want to estimate an articulated body state sequence $\phi_{1:T}$.A tracking formulation with sliding temporal window is (after Urtasun et al.):

$$\begin{aligned} p(\phi_{t:t+\tau}|I_{1:t+\tau}, M) &\propto p(I_{t:t+\tau}|\phi_{t:t+\tau})p(\phi_{t:t+\tau}|I_{1:t-1}, M) \\ &\approx \underbrace{p(I_{t:t+\tau}|\phi_{t:t+\tau})}_{\text{Image likelihood}}\ \underbrace{p(\phi_{t:t+\tau}|\phi^{MAP}_{1:t-1}, M)}_{\text{Prediction}} \end{aligned}$$

using the following notations:

- state at time t: $\phi_t = (g_t, y_t, x_t)$ with global pose g_t
- image sequence: $I_{1:T} = (I_1, ..., I_T)$
- learned GPDM: $M = \{X, \alpha, \beta, W\}$
- MAP estimate history: $\phi^{MAP}_{1:t-1}$

Image likelihood

Assuming that image measurements conditioned on states are independent, we can factorize the image likelihood

$$\underbrace{p(I_{t:t+\tau}|\phi_{t:t+\tau})}_{\text{Image likelihood}} = \prod_{i=t}^{t+\tau} p(I_i|\phi_i)$$

$$= \prod_{i=t}^{t+\tau} exp(-\frac{1}{2\sigma_e^2}\sum_{j=1}^{J}\|\hat{m}_t^j - P(p^j(\phi_t))\|^2)$$

using the following notations:

- σ_e: 10 pixels (empirical results)
- $\hat{m}_t^j$: 2D tracker image measurement of body point j
- $P(p^j(\phi_t))$: projected body point j according to model

Prediction distribution

Since the training set didn't contain global motion we factor the prediction density into a prediction over global motion, and one over poses and latent positions:

$$
\begin{aligned}
& p(\phi_{t:t+\tau} | \phi_{1:t-1}^{MAP}, M) \\
= \; & p(g_{t:t+\tau} | g_{t-2:t-1}^{MAP}) \; p(y_{t:t+\tau}, x_{t:t+\tau} | x_{t-1}^{MAP}, M) \\
= \; & \underbrace{p(g_{t:t+\tau} | g_{t-2:t-1}^{MAP})}_{\text{Global motion}} \; \underbrace{p(y_{t:t+\tau} | x_{t:t+\tau}, M)}_{\text{Pose mapping}} \; \underbrace{p(x_{t:t+\tau} | x_{t-1}^{MAP}, M)}_{\text{Dynamics}}
\end{aligned}
$$

where $M = \{X, \alpha, \beta, W\}$ denotes the learned GPDM.

Global motion

For the global rotation o_t and translation z_t a second-order Markov model is assumed

$$p(g_j|g_{j-2:j-1}) = exp(-\frac{\|z_j - \hat{z}_j)\|^2}{2\sigma_z^2} - \frac{\|o_j - \hat{o}_j)\|^2}{2\sigma_o^2})$$

where the mean prediction is

$$\hat{z}_j = 2z_{j-1} - z_{j-2} \qquad \hat{o}_j = 2o_{j-1} - o_{j-2}$$

with the initial condition at time t provided by previous MAP estimates:

$$g_{t-2} = g_{t-2}^{MAP} \qquad g_{t-1} = g_{t-1}^{MAP}$$

Pose mapping and dynamics

Assuming that a pose sequence can be factored into the density over individual poses we get:

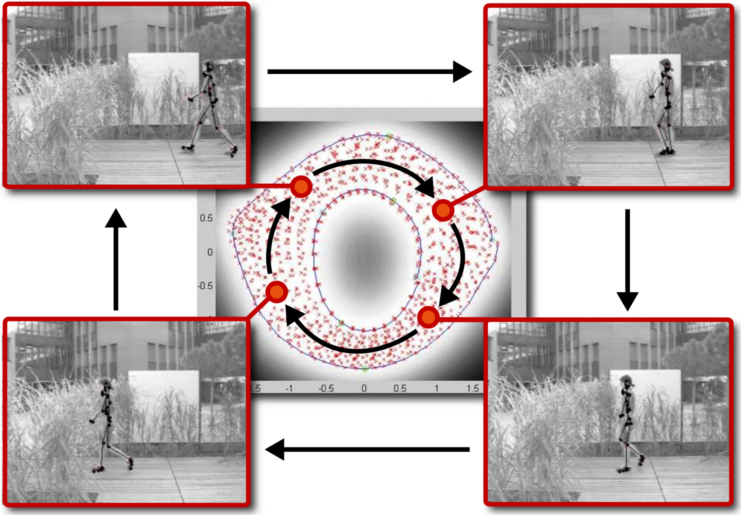
$$p(y_{t:t+\tau}|x_{t:t+\tau}, M) = \prod_{j=t}^{t+\tau} p(y_j|x_j, M)$$

Second, the dynamics

$$p(x_{t:t+\tau}|x_{t-1}^{MAP}, M)$$

is annealed because the learned GPDM dynamics often differ from the video motion.

Tracking results: Feature space and latent space



Thank you for your attention!

Any questions?

- N. D. Lawrence. *Gaussian Process Latent Variable Models for Visualization of High Dimensional Data.* NIPS, 2003.
- David J C MacKay. *Introduction to Gaussian Processes.* 1997.
- C. E. Rasmussen, C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.
- R. Urtasun, D. J.Fleet and P. Fua. *Gaussian Process Dynamical Models for 3D people tracking.* CVPR, 2006.
- J. M. Wang, D. J. Fleet, A. Hertzmann. *Gaussian Process Dynamical Models.* NIPS, 2005.