Institute of Formal and Applied Linguistics, Charles University

GAUSSIAN PROCESSES

Jan Hajič
hajicj@ufal.mff.cuni.cz

May 30, 2013

Contents

1. Introduction

We will first introduce the role of the Gaussian Process in the regression setting, then we show its definition and briefly describe its properties.

We assume the reader is familiar with Multivariate Gaussian distributions and general concepts from Bayesian inference.

1.1 Where does a Gaussian Process come from?

In a regression problem, we are looking for a function that explains the data we see and enables us to predict response values for new data. Consider a 1-d example: we would like to be able to predict real estate prices in our neighborhood, so we look at some advertisements in the local newspaper and use that as some observations for our prediction. We know real estate prices depend on how large the property is, which is readily available in each advertisement. However, using the advertisements directly, we can only estimate prices for real estate sizes we have seen in the newspaper – what if the property we're interested in is smaller than all the others? What if it's somewhere in the middle? Larger? Naturally, we want to extrapolate something from the data we already have: the general relationship between property size and price. This relationship is expressed by a function and this function is what we're trying to recover.

Note that we have already made a very strong assumption: that real estate prices only depend on the property size. In reality there are other factors at play, so we cannot hope to be able to predict prices perfectly – even if we get the best possible function that explains how prices and sizes go together, the real prices will deviate from the predictions. Because we chose not to collect and use any information about what causes these deviations, we have no choice but to regard them as random. If we could identify everything that influences property price, we could have a model with no noise (but there would never ever be enough real estate advertisements in all the world's papers to estimate that function).

This is the first set of assumptions we are making in solving a regression problem: choosing features that the response variable (in our case, property price) depends on. This is a step no clever automated method can do for us: until we tell the model what to model, it will not be able to predict anything. There are great ways of automatically selecting relevant features, but no ways of identifying the initial set of features from which to start selecting – this has to come from our knowledge of the problem (for instance, we know that property size influences the price). Fortunately, we usually do know something about the

influences on our desired response, or we can at least make a pretty good initial guess.

The second set of assumptions is about the nature of the noise. Since we know nothing or assume to know nothing about the noise, we usually assume it is Gaussian – the influences we do not know about we can view as a set of small random pertrubances and a lot of small random pertrubances sum, by the Central Limit Theorem, approximately to a Gaussian.

The third set of assumptions is about the function itself. There is a lot of functions to choose from – for instance, any straight line (aside from a vertical one). Fortunately, there is a multitude of methods to automatically choose the function that best predicts responses given our data; unfortunately, most of these methods require us to tell them at least what kind of function to expect: stepwise, piecewise linear, linear, quadratic, exponential, periodic, smooth... There is still a lot of function families to choose from, and this choice is more difficult to make based on our world knowledge than specifying the feature set: while we do know that for instance the shape of the economy does influence property prices, it is very hard to specify what the relationship this should be: a bearish market, where demand is low, pushes prices down, but a strong, bullish market may mean there is more competition; a linear approximation thus isn't always adequate (and rarely very precise). In other examples, we may wish to approximate something by a polynomial, but cannot be sure about the maximum degree of that polynomial. More generally, whenever we model something as a combination (usually linear) of basis functions (like the degrees of a polynomial), we have to make choices about how many and which basis functions to use.

The principled way of choosing the correct assumptions about the desired function is model selection: we compare the evidence of various models and choose the one which gives us the best evidence.[1] However, model selection takes a long time and we still need to hand-pick the models we will be selecting from – the point at which we must make assumptions about the function is only moved a little further back. Essentially, instead of choosing one model, we choose a pool of models, like we did with features before running feature selection.

However, model selection can be worked around. We can relax our assumptions on the function completely and only retain the intuition that *for similar points in our feature space, we should get similar responses.* When we assume the observation noise is Gaussian, a natural way of describing how good a function is as a descriptor of the relationship between the features and the response is by means of a *Gaussian Process.*

[1] This is elegantly described at `http://alumni.media.mit.edu/~tpminka/statlearn/demo/`

1.2 What is a Gaussian Process?

Let us now define a Gaussian Process (shortened as GP).

Definition 1. A Gaussian Process $\mathcal{GP}(m, k)$ with a mean function m and a positive-semidefinite covariance function k is an infinite collection of random variables, where each finite subset has a Multivariate Gaussian distribution with consistent parameters.

Given any set of points $\mathbf{x} \in \mathcal{X}$ and a function $f(x) : \mathcal{X} \to \mathbb{R}$ which is Gaussian Process-distributed, then the distribution of $f(\mathbf{x})$ is:

$$f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x})) \tag{1.1}$$

where K is a positive semi-definite matrix such that $K[i, j] = k(x_i, x_j)$.

To evaluate the probability of a GP-distributed function, we thus first have to specify some points $\mathbf{x}$ at which we evaluate the function and then use Eq. (1.1) to compute the probability of the vector $f(\mathbf{x})$. The definition of a GP tells us this vector is Multivariate Gaussian-distributed with parameters computed from $\mathbf{x}$ using the mean and covariance functions.

The consistency of GP parameters means that these functions do not depend on $\mathbf{x}$ (their values may, of course, but they will be *computed* consistently over the whole feature space $\mathcal{X}$).

This is a step from the finite-dimensional Multivariate Gaussian distributions to infinite-dimensional distributions. The GP distribution has no covariance matrix: instead, it has a covariance *function*, which can be evaluated at any point in $\mathcal{X} \times \mathcal{X}$. In the same vein, the mean vector of a Multivariate Gaussian becomes instead a mean function, which can be evaluated for any $x \in \mathcal{X}$. Since we will never be able to evaluate the function f everywhere on $\mathcal{X}$ anyway, the formulation "any set of points" in Def. 1 together with this fact already give us the power to work with random *functions*.

In the regression setting, for example, we have a set $\mathcal{D}$ of training points $\mathbf{x}$ with the associated response observations $\mathbf{y}$ and we wish to make predictions for a different set of points, $\mathbf{x}^*$. By placing a GP prior on f, without specifying anything about the form of f, we can get a posterior and a predictive distribution. (Recall the Gaussian-Gaussian conjugacy: the posterior will be a Gaussian Process as well, with a new mean and covariance function. See 2.) The information gleaned from $\mathcal{D}$ flows into updating m and k in the posterior computation.

1.3 Intuitions about GP behavior

Coming back to the intuition that we want functions predicting similar results for similar intputs, let's examine the behavior of a GP-distributed function for various $\mathbf{x}$.

The mean function merely tells us what value to expect at that given point. The covariance function, however, tells us how the points *interact*: how the value of f at point x_1 influences the value of f at a point x_2. The higher the value $k(x_1, x_2)$ is, the more are we convinced that $f(x_1)$ and $f(x_2)$ should not be very different.

A natural choice of a covariance function is one which computes $k(x_1, x_2)$ based on the distance $|x_1 - x_2|$ (these are called *stationary* covariance functions; more in 3).

Choosing the covariance function is the singularly the most important decision in using a Gaussian Process. It controls what the preferred functions should look like: how strongly do we feel that f should be very smooth, whether it can become rougher or smoother in some areas, we can specify that we want a periodic function – any concept of "similarity" of data points in our intuition which can be described by a positive semi-definite function is applicable.

2. Model, posterior and predictive

2.1 Model

Let us define the following model: we observe output generated from a function f with Gaussian noise. The function f itself is Gaussian Process-distributed. No other assumptions are made.

$$f \sim \mathcal{GP}(m, k) \tag{2.1}$$

$$y \sim \mathcal{N}(f(x), \sigma^2) \tag{2.2}$$

The mean function is denoted by m, k is the covariance function. σ^2 is the observation noise variance. We can rewrite the likelihood for a vector $\mathbf{y}$ as a Multivariate Gaussian: $\mathbf{y} \sim \mathcal{N}(f(\mathbf{x}), I_d\sigma^2)$.

Typically, m is set to a zero function. This is what a sample from a zero-mean Gaussian Process prior may look like:

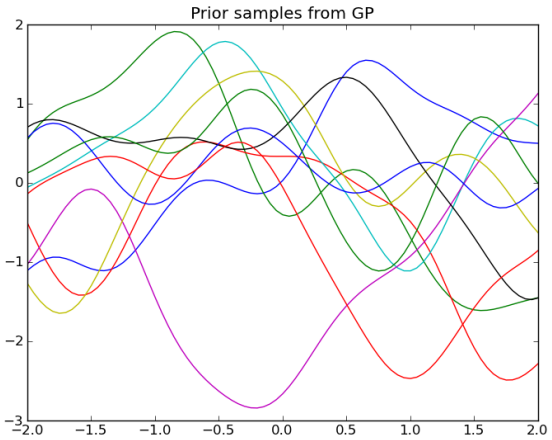


Figure 2.1: Draw from a GP prior

Given the data $\mathcal{D} = (\mathbf{x}, \mathbf{y})$, where x_i is the i-th data point and y_i is the observation in x_i, we wish to find the posterior distribution

$$P(f|\mathbf{x}, \mathbf{y}, \mathbf{x}', m, k) = P(\mathbf{y}|f, \mathbf{x})P(f|\mathbf{x}', m, k) \tag{2.3}$$

and the predictive

$$P(\mathbf{y}^*|\mathbf{x}^*, f, \mathbf{x}, \mathbf{y}, m, k) = \int_f P(\mathbf{y}^*|f, \mathbf{x}^*)P(f|\mathbf{x}, \mathbf{y}, \mathbf{x}', m, k) \tag{2.4}$$

Note the difference between $\mathbf{x}$ and $\mathbf{x}'$. $\mathbf{x}$ is the set of data points where we have measured the response, $\mathbf{x}'$ is the set of data points at which we evaluate the prior on f. (We can, of course, use the same points for both $\mathbf{x}$ and $\mathbf{x}'$.) The equations can also be written without $\mathbf{x}'$; we wanted to emphasise that the set of points at which we evaluate the *prior* probability of a function f can be different from the set of observed data points, which we use to update m and k in the *posterior*. Note that we have no way of evaluating the probability of f unless we select such a set $\mathbf{x}'$. However, the definition of a GP says that whatever set we choose to evaluate the prior on, we can compute the parameters of the Multivariate Gaussian on that set in the same way from m and k.

Herein lies the beauty of the GP prior/Gaussian likelihood model: not only do we learn something about f at points $\mathbf{x}$, we can update the *whole functions* m and k and the result is again a Gaussian Process. This is a crucial point: that under the model defined in Eq. (2.1) and Eq. (2.2), the information gained from $\mathcal{D}$ tells us something about the whole function, and this even though we do not know anything about the form of the function – we only specify constraints on more or less local behavior of f through the covariance function. (Recall the intuition that we want functions that give similar values for similar inputs.)

2.2 Posterior derivation

Now to actually show that the posterior of a GP is also a GP and how to update m and k. The trick is to bridge the mentioned gap between a finite-dimensional Multivariate Gaussian posterior on the set $\mathbf{x}$ and the updates to the complete functions m, k across all of $\mathcal{X}$. We will prove this directly from the definition of a Gaussian Process.

We have $\mathcal{D} = (\mathbf{x}, \mathbf{y})$. For any $\mathbf{x}^* \in \mathcal{X}$, (remember that this is for *any* $\mathbf{x}^*$), since f is drawn from a GP, the following holds:

$$\begin{bmatrix} f(\mathbf{x}) \\ f(\mathbf{x}^*) \end{bmatrix} \sim (N) \left(\begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{x},\mathbf{x}) & K(\mathbf{x},\mathbf{x}^*) \\ K(\mathbf{x}^*,\mathbf{x}) & K(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right) \tag{2.5}$$

This is nothing surprising: we are merely applying the definition of a GP to the combined set $\mathbf{x} \cup \mathbf{x}^*$. Note that we haven't used $\mathbf{y}$ yet. In order to do that, we must take into account the observational noise. Because it is Gaussian with a fixed variance σ^2, we can update Eq. (2.5) as:

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{x}^*) \end{bmatrix} \sim (N) \left(\begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{x},\mathbf{x}) + \mathbf{I}\sigma^2 & K(\mathbf{x},\mathbf{x}^*) \\ K(\mathbf{x}^*,\mathbf{x}) & K(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right) \tag{2.6}$$

The expression $K(\mathbf{x},\mathbf{x}) + \mathbf{I}\sigma^2$ reflects this shift from the probability distribution of the function values on $\mathbf{x} \cup \mathbf{x}^*$ to the distribution of the combined vector $(\mathbf{y}, \mathbf{x}^*)$. The additional variance follows from the form of the observational noise, using the formula for variance of compound Gaussians.

Recall the conditional property of Multivariate Gaussians: given that

$$\begin{bmatrix} f(\mathbf{x_1}) \\ f(\mathbf{x_2}) \end{bmatrix} \sim (N) \left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{2.7}$$

the conditional $f_2|f_1$ takes the form

$$f_1|f_2 \sim \mathcal{N}(\beta_0 + \beta_1 f_1, \Sigma_{2|1}) \tag{2.8}$$

where

$$\beta_0 = \mu_2 - \Sigma_{21}\Sigma_{11}^{-1}\mu_1 \tag{2.9}$$

$$\beta_1 = \Sigma_{21}\Sigma_{11}^{-1} \tag{2.10}$$

$$\Sigma_{2|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} \tag{2.11}$$

Plugging Eq. (2.8) into Eq. (2.6), we obtain the expression for the posterior on $f(\mathbf{x}^*)$:

$$f(\mathbf{x}^*) \sim \mathcal{N}(\hat{m}(\mathbf{x}*), \hat{K}(\mathbf{x}*, \mathbf{x}*)) \tag{2.12}$$

where

$$\hat{m}(\mathbf{x}^*) = m(\mathbf{x}^*) + K(\mathbf{x}^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{2.13}$$

$$\hat{K}(\mathbf{x}^*, \mathbf{x}^*) = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}K(\mathbf{x}, \mathbf{x}^*) \tag{2.14}$$

Note the dimensionality of the various variables. Suppose that $\mathbf{x}$ is a vector of n dimensions, $\mathbf{x}^*$ of n^* dimensions. Then:

- $m(\mathbf{x}^*)$ is an n^*-dimensional vector, where item i has value $m(x_i^*)$,
- $K(\mathbf{x}^*, \mathbf{x})$ is an $n^* \times n$-dimensional matrix, where item i, j has value $k(x_i^*, x_j)$,
- $(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)$ is an $n \times n$-dimensional matrix, where item i, j has value $k(x_i, x_j) + \sigma^2\mathbf{1}(i = j)$,
- $(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}$ is its inverse,
- $\mathbf{y} - m(\mathbf{x})$ is an n-dimensional vector.

The generalization from a posterior on $f(\mathbf{x}^*)$ to the Gaussian Process posterior follows from the definition: since we have shown that Eq. (2.12) holds for any $\mathbf{x}^* \in \mathcal{X}$, the posterior is a Gaussian Process and the posterior mean and covariance functions are

$$f(x^*) \sim \mathcal{N}(\hat{m}(x*), \hat{k}(x*, x*)) \tag{2.15}$$

where

$$\hat{m}(x^*) = m(x^*) + K(x^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{2.16}$$

$$\hat{k}(x_i^*, x_j^*) = k(x_i^*, x_j^*) - k(x_i^*, \mathbf{x})(k(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}k(\mathbf{x}, x_j^*) \tag{2.17}$$

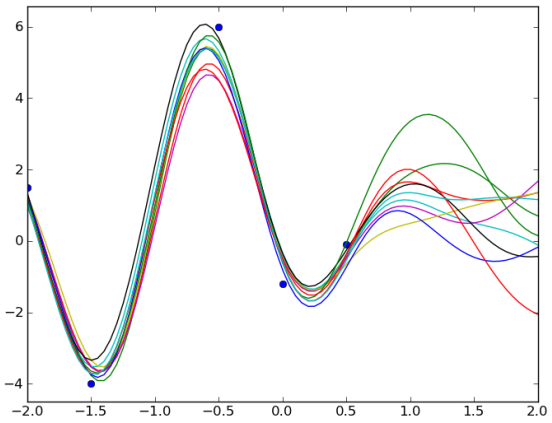


Figure 2.2: Draw from a GP posterior

This is a sample from a GP posterior. The observations are rendered as blue dots. Note that the posterior samples do not pass through more extreme values of y. This is due to the role of the observation noise, modeled as likelihood variance σ^2. See section 2.4.

Notice how areas further from the observed points are quickly becoming uncertain.

2.3 Predictive

The predictive distribution $p(y^*|x^*, \mathbf{x}, \mathbf{y})$ is easily computed as:

$$\begin{aligned} p(y^*|x^*, \mathbf{x}, \mathbf{y}) &= \int p(y^*|f(x^*))p(f(x^*)|\mathbf{x}, \mathbf{y})df(x^*) \\ &= \int \mathcal{N}(y^*|f(x^*), \sigma^2)\mathcal{N}(f(x^*)|\mathbf{y}, \mathbf{x})df(x^*) \end{aligned} \tag{2.18}$$

which leads to a Gaussian predictive distribution (mean is posterior mean, variance follows from compounding Gaussians this way, as in Eq. (2.6)):

$$y^*|x^*, \mathbf{x}, \mathbf{y} \sim \mathcal{N}(\hat{m}(x^*), \hat{k}(x^*, x^*) + \sigma^2) \tag{2.19}$$

2.4 Interpretation of posterior mean and covariance

To begin dissecting what the individual expressions actually *mean*, we'll temporarily reduce $\mathbf{x}^*$ to a scalar and set $m(x) = 0$ for each $x \in \mathcal{X}$. Then, the expression for the posterior mean $\hat{m}(x^*)$ is

$$\hat{m}(x^*) = k(x^*, \mathbf{x})^T (K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}\mathbf{y} \tag{2.20}$$

The first member, $k(\mathbf{x}^*, \mathbf{x})$, computes the covariance of x^* with each point in $\mathbf{x}$. This gives us an n-dimensional vector. We transpose it, so that it's "horizontal". Let's leave the inverse in the middle out for a moment (pretend it's an identity matrix). What happens next? We multiply $k(\mathbf{x}^*, \mathbf{x})^T$ by $\mathbf{y}$, which is the dot product of two n-dimensional vectors: thus we get a scalar prediction of $f(x^*)$.

We can look on this operation as taking a sort of weighted average over all the y_i's. So, what $\hat{m}(x^*)$ does is that it predicts the mean of f at x^* to be a certain average of the values of f in the various x_i's, weighted by how similar we expect the value of $f(x^*)$ to be to each x_i.

What is the role of the middle term, the modified covariance inverse?

Imagine that there are 10 points in $\mathbf{x}$, all the points in $\mathbf{x}$ are close to each other and that the value of $f(x)$ in the region is about 10 or so. If x^* is close to $\mathbf{x}$, with $k(x^*, x_i)$ approaching 1, the weighted average of all the y_i's will be approaching 100, which is definitely not right – we want the result to be, again, about 10.

This would be the case if we only took one point from $\mathbf{x}$. Since $\mathbf{x}$ is so densely packed, taking more points into account does not give us a lot of new information about the value of $f(x)$ in their vicinity. The closer the points are, the less informative adding new points becomes and the higher their covariance is. (We're talking about "closeness" in the sense of "high covariance" only, k is not necessarily a metric function!) So, the inverse of the covariance is used to normalize the "weights" of individual y_i's.

Again, as the "weighted average" notion was just an intuition, the normalization is also just an intuitive concept: it does not necessarily mean the vector $k(x^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \mathbf{I}\sigma^2)^{-1}$ will sum to 1. What the operation really does is that it accounts for how informative the *combined* contributions of points in $\mathbf{x}$ are: if two people in the same building yell that there is a fire, we kind of assume that it's the same fire, whereas if each of them come yelling from a different city block,

it's probably two fires.[1] This transformation of the covariance vector $k(x^*, \mathbf{x})$ is not hierarchical – the "information gain"[2] from each point in a highly co-varied group is tempered in accordance to how much would we lose if we removed this point from $\mathbf{x}$, so each of the pseudo-weights in our example of 10 closely co-varied points should be tempered to about 1/10. (Not exactly, unless we are in a space where all the points are the same "distance" from each other as well as the point x^*.)

Looking again at Eq. (**??**):

$$\hat{m}(x^*) = k(x^*, \mathbf{x})^T (K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}\mathbf{y} \tag{2.21}$$

we see that to estimate $f(x^*)$, the new information we gain from $\mathcal{D}$ is used thus: the point x^* gets "transformed" to a similarity space, where the individual points in $\mathbf{x}$ each are direction in this space. The $\mathbf{y}$ term tells us the value of $f(x)$ we should expect at similarity 1 to the direction x_i. The modified inverse covariance matrix tells us how much the directions agree on their prediction only because they point in the same direction, i.e. how informative the directions are.

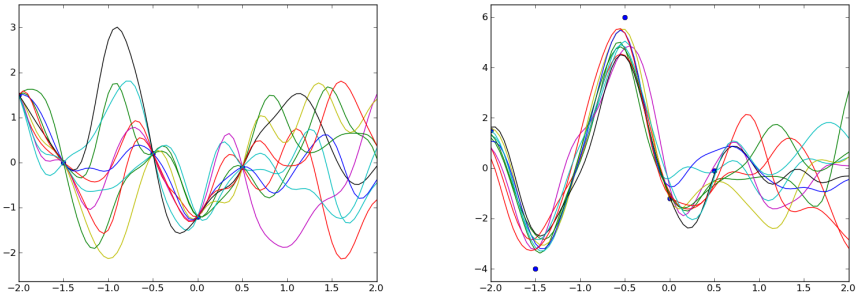
Finally, what is the role of σ^2? Imagine the covariance matrix is diagonal. The higher the likelihood variance, the smaller the values on the diagonal on the inverse. This is a coefficient in the "new information" term: with growing σ^2, $\mathbf{y}$ is a less valuable information on $f(\mathbf{x})$, so we leave more influence to our prior belief $\hat{m}(x)$.

If the similarity space were metric, all these intuitions are easy to see. However, the only limitation on the covariance function is that it must be positive semi-definite, which can cloud these intuitions heavily. More on covariance functions in chapter 3.

The step from Eq. (2.15) to Eq. (2.13) is easy: we only relax the assumption that $m(x) = 0$ for all $x \in \mathcal{X}$ and the expected value for $\hat{m}(x^*)$ is thus the original

[1] To continue this rather grim example: Fire has, at the time people come yelling about it, a covariance function that quickly decreases with distance, so if our prior belief is that the city is not on fire, chances of fire *between* the city blocks are rather meagre. However, with time, fire spreads, so its covariance function does not decrease so quickly with distance from the data points and the chances of something in the middle of the two city blocks burning goes up significantly. This is reflected in the first term, $k(x^*, \mathbf{x})$ – for x^* in between the burning city blocks, at the start of the fire, the covariance with either burning block is low; as the fire spreads, the covariance changes and the chances of being on fire go up significantly. However, after enough time (let's hope it doesn't come to that!), the chances of one of the fires spreading all the way to the source of the other becomes so high that knowing the other place is burning as well doesn't really tell us anything new; this will be reflected in the decreasing inverse covariance, in the pseudo-normalization term.)

[2] In quotes, because it shouldn't be confused with the information-theoretic measure of the same name!



(a) Posterior, noise-free GP (b) Posterior, noisy GP ($\sigma^2 = 0.25$)

Figure 2.3: Influence of noise on posterior

expectation for x^* modified by how much this prior expectation differs from what we've actually seen in $\mathcal{D}$ (the term $\mathbf{y} - m(\mathbf{x})$). We see that this formulation gives us the ability to continuously update $\hat{m}$ with new data.

What happens to $\hat{K}$? Recall Eq. (2.14):

$$\hat{K}(\mathbf{x}^*, \mathbf{x}^*) = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}K(\mathbf{x}, \mathbf{x}^*) \tag{2.22}$$

The first term is the original covariance k applied to x^*. The second term looks, in principle, rather similar to the transformation from m to $\hat{m}$: we take x^*, compute its covariance with all $x_i \in \mathbf{x}$, normalize by $(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}$ and transform back into the x^*-space. Both terms are positive, so we are *decreasing* the original covariance for x^*.

What is happening in the negative term? Let's focus on one cell i, j in the covariance matrix $\hat{K}(\mathbf{x}^*, \mathbf{x}^*)$.

$$\hat{k}(x_i^*, x_j^*) = k(x_i^*, x_j^*) - K(x_i^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + I\sigma^2)^{-1}K(\mathbf{x}, x_j^*) \tag{2.23}$$

Forgetting the pseudo-normalization for a moment, as we've done with the mean, we see a dot product of how similar[3] x_i^* is to points in $\mathbf{x}$ and how similar x_j^* to these points. In the "weighed average" view, we are weighing (pseudo-weighing) the similarities of $\mathbf{x}$ to x_j^* by the similarities of x_i^* to $\mathbf{x}$. We are computing something like the cosine similarity of x_i^* and x_j^* in the $\mathbf{x}$-space. If they are co-varied with the same points in $\mathbf{x}$, a lot is known about what their values should be, so the posterior should not prefer $f(\mathbf{x}^*)$ that make x_i^* and x_j^* dependent on each other that if one

[3] "Similar" means co-varied here.

Essentially, the subtraction tells us how much the information about x_i^* given by x_j^* can be explained away by simply watching how similar they are to points in $\mathbf{x}$. This reflects the fact that after seeing $\mathbf{x}$ and $\mathbf{y}$, the points in $\mathbf{x}^*$ are not the sole source of information on each other. Generally speaking, if a point x_i^* is similar to some x_i from $\mathbf{x}$, where the value of f has been observed, then various values of the random variable $f(x_j^*)$ don't really imply large changes in the distribution of $f(x_i^*)$, which will tend to stay closer to $f(x_i)$. If x_i^* is identical to x_i and the process is noise-free (σ^2 is 0), then the covariance of any point in x^* to any point in $\mathbf{x}$ is zero: we have already observed $f(x_i^*)$, what's going on at x_j^* will not make us change our minds. Another scenairo where the information gain on covariance from the data has very pronounced effects is the following:

- x_i^* and x_j^* are both significantly closer to some x than to each other,
- the likelihood variance σ^2 is very low,
- the prior variance $k(x, x)$ is reasonably high,
- other points from $\mathbf{x}$ are not relevant.

Then the term $k(x_i^*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \mathbf{I}\sigma^2)^{-1}k(\mathbf{x}, x_j^*)$ becomes larger than $k(x_i^*, x_j^*)$ and the posterior covariance is negative. This makes good sense: the covariance function is responsible for keeping f smooth, so in a certain small area around x, we strongly want to avoid ex treme shapes of the function. So, if $f(x_i^*)$ grows, since $f(x)$ has been observed with great reliability (low likelihood variance), in order for the function to remain smooth, $f(x_j^*)$ on the other side of x will preferrably decrease.

Again, if σ^2 is higher, the inverse is smaller, so the influence of the covariance with $\mathbf{x}$ is not so informative: the values of $\mathbf{y}$ are less certain to represent what the true value of $f(x)$ is at that point and the prior $k(x_1^*, x_2^*)$ retains more influence.

3. Covariance functions

While the choice of the prior mean does not influence the behavior of f much, the choice of the covariance function is crucial. In the explanation of how the posterior mean is constructed and what information it encodes, we have seen that the covariance function defines what we consider similar and how the notion of similarity behaves.

There are many covariance functions to choose from; the only restriction is that they be positive semi-definite. The most commonly used family of covariance functions is *stationary*: these functions depend only on $|x - x'|$, not on where in $\mathcal{X}$ the points x and x' lie.[1]

Covariance functions are generally built to prefer smooth functions with no sudden changes. They fulfill the role of reducing model complexity and thus prevent overfitting. Most covariance functions that are commonly used have an associated hyperparameter which controls how strongly they resist "rough" functions (large values of f'').

Our preferred way of thinking about stationary covariance functions is to imagine that each data point x has a certain "sphere of influence" where the posterior prefers values similar to $f(x)$. This sphere of influence can be visualized by plotting $k(x, x')$ for a fixed x. One natural choice of a stationary covariance function is a bell curve; the intuition is then quite similar to how a Support Vector Machine with radial basis kernel function works.[2] There are various other covariance functions with very nice interpretations.

In the following sections, we give examples of common covariance functions.

3.1 Squared exponential covariance function

The squared exponential covariance function has the following form:

$$k_{SE}(x, x') = v_0 \exp\left(-\frac{1}{2l^2}(x - x')^2\right) \tag{3.1}$$

It is stationary and infinitely differentiable. Notice the two hyperparameters, v_0 and l.

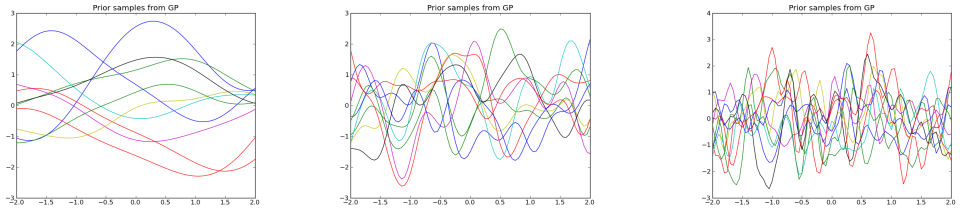
[1] Some authors refer to these functions as *stationary and isotropic* and define stationary functions as depending only on $x - x'$.

[2] In fact, we think SVM operation with RBF kernels could perhaps be formulated using a Gaussian Process.

The l hyperparameter is called the *length scale*, since it controls how quickly a data point's influence will scale, or in other words, how "long" the "waves" of $f(x)$ will be – how prone to changing value quickly f is. The higher l is, the smaller the exponentiated term for constant $|x - x'|$, so the slower the influence of x fades as we move away from it.

The v_0 hyperparameter controls pointwise prior variability: if $x = x'$, then $k_{SE}(x, x') = v_0$. The larger v_0 is, the higher deviations from $m(x)$ we can expect. (l, on the contrary, only controls how long it takes for the function to "turn around".) A draw from a GP with low v_0 and l will oscillate unpredictably in a region close to $m(x)$, a draw from GP with high v_0 and l will look like long, massive waves.

Examples of a GP prior with the squared exponential covariance function, various length scales:



(a) l = 1 (b) l = 0.25 (c) l = 0.1

Figure 3.1: SE covariance function, various length scales

The squared exponential covariance function is a natural choice of a covariance function for many more reasons than just its ability to manipulate the length scale (other covariance functions can achieve much finer control). We will show in 4.2 that it is intimately connected to linear models with basis functions that imitate the notion of general similarity to a point in $\mathcal{X}$.

3.2 Rational quadratic covariance function

The rational quadratic covariance function has the following form:

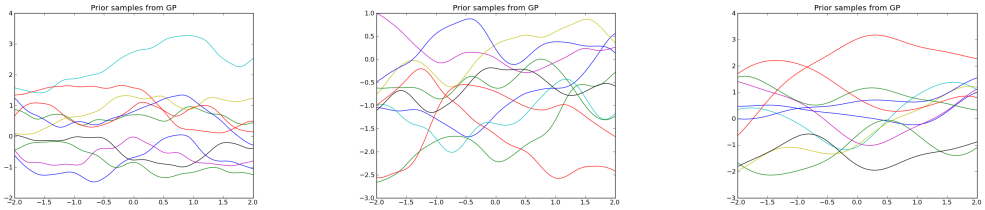
$$k_{RQ}(x, x') = \left(1 + \frac{(x - x')^2}{2\alpha l^2}\right)^{-\alpha} \tag{3.2}$$

There are two hyperparameters, $\alpha > 0$ and $l > 0$. The rational quadratic covariance function can be interpreted as a mixture of squared exponential functions with different length scales. Let $\tau = -l^2$, $d = (x - x')$, $\beta = l^{-2}$ and assume $\tau \sim \mathbf{Gam}(\alpha, \alpha/\beta)$. Then:

$$
\begin{aligned}
k_{RQ}(d) &= \int k_{SE}(d|\tau)p(\tau|\alpha,\beta)d\tau \\
&\propto \int \exp\left(-\frac{1}{2}\tau d^2\right)\tau^{\alpha-1}\exp\left(-\frac{\alpha}{\beta}\right)d\tau \\
&\propto \left(1+\frac{(x-x')^2}{2\alpha l^2}\right)^{-\alpha}
\end{aligned}
\tag{3.3}
$$

The α hyperparameter is the shape parameter of the Gamma distribution: the higher α is, the more weight is given to the k_{SE}s with higher l. The β hyperparameter is a scale parameter: the higher β is, the more "spread out" the mixture of k_{SE}s becomes.

Examples of a GP prior with the rational quadratic covariance function, various length scales:



(a) l = 1 (b) l = 0.25 (c) l = 0.1

Figure 3.2: RQ covariance function, various length scales

TODO: Matern covariance functions, periodic covariance functions

4. Relationship with linear models

There is an intimate connection between Gaussian Processes and linear models with a Gaussian prior on weights that makes GP even more interesting than they have seemed so far. In this chapter, which is not necessary for understanding what a GP is and how it works, we will explore this connection.

4.1 Finite linear models

Let us first define such a model:

Definition 2. A finite linear model (FLM) with a Gaussian prior on the weights is a function

$$f(x) = \sum_{m=1}^{M} w_m \phi_m(x), \qquad w \sim \mathcal{N}(0, \Sigma_0) \tag{4.1}$$

The ϕ_ms are called either *factors*, or *basis functions*. We will use the latter. (This model is linear in the sense that it is a linear combination of the basis function values for a given x, not necessarily of the features in x. The basic linear model would have the form $f(x) = x^T \mathbf{w}$. Notice that the dimensionality of the vector $\mathbf{w}$ is also different: in the basic linear model, there are as many weights as there are features, in our FLM, there are as many weights as there are basis functions.)

Thus, for any $\mathbf{x} = x_1, \ldots, x_n$, the joint distribution of $(f(x_1), \ldots, f(x_n))$ is Multivariate Gaussian: $\phi_m(x)$ are fixed coefficients and the "moveable" part, $\mathbf{w}$, is distributed according to a Multivariate Gaussian. Therefore, f is distributed according to some Gaussian Process.

What are the parameters of this mystery GP? The mean function will be, using simply the definition of expected value,

$$\begin{aligned} m(x) &= \mathrm{E}_w \left[f(x) \right] \\ &= \sum_{m=1}^{M} \mathrm{E}_w \left[w_m \right] \phi_m(x) \\ &= 0 \end{aligned} \tag{4.2}$$

and the covariance function is, from the definition of covariance,

$$\begin{aligned} k(x,x') &= \mathbf{E}_w\left[f(x)f(x')\right] - \underbrace{\mathbf{E}_w\left[f(x)\right]\mathbf{E}_w\left[f(x')\right]}_{=0 \text{ (see Eq. (4.2))}} \\ &= \mathbf{E}_w\left[\sum_{m=1}^{M}\sum_{m'=1}^{M} w_m w_{m'}\phi_m(x)\phi_{m'}(x')\right] \\ &= \sum_{m=1}^{M}\sum_{m'=1}^{M} \underbrace{\mathbf{E}_w\left[w_m w_{m'}\right]}_{\Sigma_{0m,m'}}\phi_m(x)\phi_{m'}(x') \\ &= \phi(x)^T\Sigma_0\phi(x') \end{aligned} \tag{4.3}$$

Note that $\phi_m(x)$ is a scalar, while $\phi(x)$ is an M-valued vector.

This equation reflects the role of $\phi_m(x)$ in Eq. (4.1), which seemed a little forgotten so far. If we saw large values of $phi_m(x)$ for both x_i and x_j, even though the prior covariance of the corresponding weights w_i, w_j might not be very high, because the factors agreed on what kind of value, to give to $\mathbf{x}$, the sum over all factors for x and x' might be a bit more similar than chance.

4.2 Infinite linear models

Eq. (4.2) and Eq. (4.3) are an interesting enough result: as long as $\phi(x)^T\Sigma_0\phi(x')$ is positive semi-definite, a finite linear model with any amount of any basis functions and a Gaussian prior on weights corresponds to some GP.

However, building such a model still leaves us with decisions about the numbe of ϕs, which is the type of decisions we tried to get rid of by using a GP in the first place. We can go one step further and take the limit as m goes to infinity:

$$\begin{aligned} f(x) &= \lim_{m\to\infty}\frac{1}{M}\sum_m w_m(x)\phi_m(x) \\ &= \int_{-\infty}^{\infty} w(u)\phi_m(x)du \end{aligned} \tag{4.4}$$

The prior on weights is set to $w_m \sim \mathcal{N}(0,1)$. The weight vector becomes a weight function, thus instead of w_m we have $w(u)$. (The fraction $1/M$ in the limit has a normalizing role – if we left it out, $f(x)$ would run off into infinity. In FLMs, this role of keeping the estimate of $f(x)$ from running wild falls to the weights or is irrelevant, but in the infinite case, the weights would all have to tend to 0.

This way, the weights can be M times larger, so with increasing M, we retain the weights in a form in which the number of basis functions doesn't matter.)

The question is, again, what the mean and covariance functions of this GP are. From Eq. (4.4) and Eq. (4.2), the mean is:

$$\begin{aligned} m(x) &= \mathbf{E}_w \left[f(x) \right] \\ &= \int_{-\infty}^{\infty} \underbrace{\mathbf{E}_w \left[w(u) \right]}_{=0} \phi_u(x) du \\ &= 0 \end{aligned} \tag{4.5}$$

The general form of the covariance function is, from Eq. (4.5), since Σ_0 is an identity matrix ($w_u \sim \mathcal{N}(0, 1)$):

$$\begin{aligned} k(x, x') &= \lim_{m \to \infty} \frac{1}{M} \sum_{m=1}^{M} \phi_m(x) \phi_m(x') \\ &= \int_{-infty}^{infty} \phi_u(x) \phi_u(x') du \end{aligned} \tag{4.6}$$

We promised in 3.1 we would show why the squared exponential covariance function is a natural choice. Let us consider basis functions that are Gaussian-shaped and uniformly distributed over $mathcalX$:

$$\phi_m(x) = \exp(\frac{-(x - \frac{m}{M})^2}{2l^2}) \tag{4.7}$$

What is the point of choosing such basis functions? They tie back to the intuition of "similar values for similar points" – a point $x \in \mathcal{X}$ is transformed to a vector of how similar it is to other points in $\mathcal{X}$, specifically those on which a basis function is centered. (This is suspiciously similar to what we did in Eq. (2.20), transforming x^* to a "similarity space" to the points we saw in the data; we'll tie this loose end later.) Plus, from the Central Limit Theorem, this choice of basis functions can be interpreted thus: given a lot of small random pertrubations we know nothing about, what's the chance that when looking at x, we're actually looking at some m?

The mean of the resulting GP is again 0. From Eq. (4.6), the covariance is computed as:

$$
\begin{aligned}
k(x, x') &= \int_{-\infty}^{\infty} \exp(-\frac{(x-u)^2}{2l^2})\exp(-\frac{(x'-u)^2}{2l^2}) \\
&= ...
\end{aligned}
\tag{4.8}
$$

TODO: finish this integral, write up on Mercer's Theorem and infinite basis function expansion

List of Tables