Gaussian Processes and Relevance Vector Regression

Chapter 8, Prince Book
Goal for Today

• Sketchy introduction to Gaussian Processes and Relevance Vector Regression, so that when we see these terms in our readings, we have a rough idea what they are.

• This is NOT a rigorous introduction!
Recall what we have covered

• Linear regression
• Bayesian solution
• Non-linear regression
• Kernel Trick
Review: Linear Regression

We have one equation for each \( x, w \) training pair:

\[
Pr(w_i | x_i, \theta) = \text{Norm}_{w_i} \left[ \phi^T x_i, \sigma^2 \right]
\]

Joint Likelihood over whole training dataset

\[
Pr(w | X) = \text{Norm}_w [X^T \phi, \sigma^2 I]
\]

where

\[
X = [x_1, x_2 \ldots x_N] \quad w = [w_1, w_2, \ldots, w_N]^T
\]
Linear Regression

Maximum likelihood

\[ \hat{\theta} = \arg\max_{\theta} [Pr(w|X, \theta)] = \arg\max_{\theta} [\log Pr(w|X, \theta)] \]

Substituting in

\[ \hat{\phi}, \hat{\sigma}^2 = \arg\max_{\phi, \sigma^2} \left[ -\frac{N \log[2\pi]}{2} - \frac{N \log[\sigma^2]}{2} - \frac{(w - X^T \phi)^T (w - X^T \phi)}{2\sigma^2} \right] \]

Take derivative, set result to zero and re-arrange:

\[ \hat{\phi} = (XX^T)^{-1}Xw \]
\[ \hat{\sigma}^2 = \frac{(w - X^T \phi)^T (w - X^T \phi)}{N} \]
Review: Bayesian Regression

Likelihood

\[ Pr(w|X) = \text{Norm}_w[X^T \phi, \sigma^2 I] \]

Prior

\[ Pr(\phi) = \text{Norm}_\phi[0, \sigma_p^2 I] \]

encourages values of \( \phi \) to be small

Bayes rule’

\[ Pr(\phi|X, w) = \frac{Pr(w|X, \phi)Pr(\phi)}{Pr(w|X)} \]
Bayesian Regression

\[ Pr(\phi|X, w) = \text{Norm}_\phi \left[ \frac{1}{\sigma^2} A^{-1} Xw, A^{-1} \right] \]

where

\[ A = \frac{1}{\sigma^2} XX^T + \frac{1}{\sigma_p^2} I \]
Bayesian Regression

\[ Pr(w^*|x^*, X, w) = \int Pr(w^*|x^*, \phi) Pr(\phi|X, w) d\phi \]

\[ = \int \text{Norm}_{w^*}[\phi^T x^*, \sigma^2] \text{Norm}_\phi \left[ \frac{1}{\sigma^2} A^{-1} X w, A^{-1} \right] d\phi \]

\[ = \text{Norm}_{w^*} \left[ \frac{1}{\sigma^2} x^{*T} A^{-1} X w, x^{*T} A^{-1} x^* + \sigma^2 \right] . \]
Review: Non-linear regression

Linear regression:

\[
Pr(w_i | x_i, \theta) = \text{Norm}_{w_i} \left[ \phi^T x_i, \sigma^2 \right]
\]

Non-Linear regression:

\[
Pr(w_i | x_i, \theta) = \text{Norm}_{w_i} \left[ \phi^T z_i, \sigma^2 \right]
\]

where \( z_i = f[x_i] \)

In other words, create \( z \) by evaluating \( x \) against basis functions, then linearly regress against \( z \).
Example: polynomial regression

\[ Pr(w_i|x_i) = \text{Norm}_{w_i}[\phi_0 + \phi_1 x_i + \phi_2 x_i^2 + \phi_3 x_i^3, \sigma^2] \]

A special case of

\[ Pr(w_i|x_i) = \text{Norm}_{w_i}[\phi^T z_i, \sigma^2] \]

Where

\[ z_i = \begin{bmatrix} 1 \\ x_i \\ x_i^2 \\ x_i^3 \end{bmatrix} \]
Radial basis functions

\[ z_i = \begin{bmatrix} 1 \\ \exp \left[ -(x_i - \alpha_1)^2 / \lambda \right] \\ \exp \left[ -(x_i - \alpha_2)^2 / \lambda \right] \\ \exp \left[ -(x_i - \alpha_3)^2 / \lambda \right] \\ \exp \left[ -(x_i - \alpha_4)^2 / \lambda \right] \\ \exp \left[ -(x_i - \alpha_5)^2 / \lambda \right] \\ \exp \left[ -(x_i - \alpha_6)^2 / \lambda \right] \end{bmatrix} \]
Review: Nonlinear Regression

Maximum likelihood estimation for nonlinear regression is the same as linear regression, but substitute in $Z$ for $X$:

$$\hat{\phi} = (ZZ^T)^{-1}Zw$$

$$\hat{\sigma}^2 = \frac{(w - Z^T \phi)^T(w - Z^T \phi)}{N}$$
Bayesian Nonlinear Regression

Bayesian nonlinear regression likewise proceeds similarly to Bayesian linear regression, using features $Z$ instead of original data $X$

Through suitable rewriting of the matrix inverse equations, we can write the predictive distribution as

$$Pr(w^* | z^*, X, w) =$$

$$\text{Norm}_w \left[ \frac{\sigma_p^2}{\sigma^2} z^* T Z w - \frac{\sigma_p^2}{\sigma^2} z^* T Z \left( Z^T Z + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} Z^T Z w, \right.$$

$$\left. \sigma_p^2 z^* T z^* - \sigma_p^2 z^* T Z \left( Z^T Z + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} Z^T z^* + \sigma^2 \right]$$
Review: The Kernel Trick

Notice that the final equation doesn’t need the data itself, but just dot products between data items of the form $z_i^T z_j$

\[
Pr(w^*|z^*, X, w) = \operatorname{Norm}_w \left[ \frac{\sigma_p^2}{\sigma_2^2} z^*^T Z w - \frac{\sigma_2^2}{\sigma_p^2} z^*^T Z \left( Z^T Z + \frac{\sigma_2^2}{\sigma_p^2} I \right)^{-1} Z^T Z w, 
\sigma_p^2 z^*^T z^* - \sigma_p^2 z^*^T Z \left( Z^T Z + \frac{\sigma_2^2}{\sigma_p^2} I \right)^{-1} Z^T z^* + \sigma_2^2 \right]
\]

So, we take data $x_i$ and $x_j$ pass through non-linear function to create $z_i$ and $z_j$ and then take dot products of different $z_i^T z_j$
The Kernel Trick

So, we take data $x_i$ and $x_j$ pass through non-linear function to create $z_i$ and $z_j$ and then take dot products of different $z_i^T z_j$

Key idea:

Define a “kernel” function that does all of this together.
- Takes data $x_i$ and $x_j$
- Returns a value for dot product $z_i^T z_j$

If we choose this function carefully, then it will correspond to some underlying $z = f[x]$.

Never compute $z$ explicitly - can be very high or infinite dimension
Example Kernels

- **linear** $k[x_i, x_j] = x_i^T x_j,$

- **degree $p$ polynomial** $k[x_i, x_j] = (x_i^T x_j + 1)^p,$

- **radial basis function (RBF) or Gaussian**

  $k[x_i, x_j] = \exp \left[ -0.5 \left( \frac{(x_i - x_j)^T (x_i - x_j)}{\lambda^2} \right) \right].$

(Equivalent to having an infinite number of radial basis functions at every position in space. Wow!)
Gaussian Process Regression

- Bayesian nonlinear regression using kernels!
Gaussian Process Regression

Bayesian nonlinear regression

\[
Pr(w^* | z^*, X, w) =
\left[ \frac{\sigma_p^2}{\sigma^2} z_*^T Z w - \frac{\sigma_p^2}{\sigma^2} z_*^T Z \left( Z^T Z + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} Z^T Z w, \right.
\]

\[
\left. \sigma_p^2 z_*^T z^* - \sigma_p^2 z_*^T Z \left( Z^T Z + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} Z^T z^* + \sigma^2 \right]
\]

Gaussian Process regression

\[
Pr(w^* | x^*, X, w) =
\left[ \frac{\sigma_p^2}{\sigma^2} K[x^*, X] w - \frac{\sigma_p^2}{\sigma^2} K[x^*, X] \left( K[X, X] + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} K[X, X] w, \right.
\]

\[
\left. \sigma_p^2 K[x^*, x^*] - \sigma_p^2 K[x^*, X] \left( K[X, X] + \frac{\sigma^2}{\sigma_p^2} I \right)^{-1} K[X, x^*] + \sigma^2 \right]
\]

where the notation \( K[X, X] \) represents a matrix of dot products where element \((i, j)\) is given by \( k[x_i, x_j] \).
Regression Model Roadmap

- Linear regression
  - Problem 1: Overconfident
  - Bayesian formulation
- Problem 2: Linear
  - Project data through non-linearity
- Problem 3: Overlearning & computational cost
  - Sparse prior
- Bayesian linear regression
- Non-linear regression
- Sparse linear regression
- Gaussian process regression
- Relevance vector regression
Sparse Linear Regression

Perhaps not every dimension of the data $\mathbf{x}$ is informative.

A sparse solution forces some of the coefficients in $\phi$ to be zero.

Method:

– apply a different prior on $\phi$ that encourages sparsity

– product of t-distributions

probability mass tends to cluster around the coordinate axes, encouraging sparseness.
Sparse Linear Regression

Unfortunately, a t-distribution prior is not conjugate to the Gaussian likelihood function, and we can no longer compute the posterior / predictive distributions in closed form.

Have to resort to computing an approximation.
Sparse Linear Regression

regular linear regression computes general $\phi$ vector

in sparse regression, small components of the $\phi$ vector tend to become 0
Unfortunately, this approach becomes intractable for the non-linear regression when we have a large, high dimensional feature space. To solve this problem, we must move to a dual model.
Dual Linear Regression

KEY IDEA: rewrite gradient as a weighted combination of data points

Gradient $\Phi$ is just a vector in the data space

Can represent as a weighted sum of the data points

$$\phi = X\psi$$

Now solve for $\Psi$. One parameter per training example (rather than one parameter per feature space dimension).
Dual Linear Regression

KEY IDEA: rewrite gradient as a weighted combination of data points

Note, when dimensionality of feature space $D$ is greater than the number of training examples $N$, then $X\psi$ can only span a subspace of possible gradient vectors $\phi$.

This is OK though, because the examples $X$ exhibit no variation along those directions in feature space.
Dual Linear Regression

Original linear regression:

$$ Pr(w|X) = \text{Norm}_w[X^T \phi, \sigma^2 I] $$

Dual variables:

$$ \phi = X\psi $$

Dual linear regression:

$$ Pr(w|X, \theta) = \text{Norm}_w [X^T X\psi, \sigma^2 I] $$
Maximum likelihood

Maximum likelihood solution:

\[
\hat{\psi}, \hat{\sigma}^2 = \arg \max_{\psi, \sigma^2} \left[ -\frac{N \log[2\pi]}{2} - \frac{N \log[\sigma]}{2} - \frac{(w - X^T X \psi)^T (w - X^T X \psi)}{2\sigma^2} \right]
\]

Dual variables:

\[
\hat{\psi} = (X^T X)^{-1} w
\]

\[
\hat{\sigma}^2 = \frac{(w - X^T X \psi)^T (w - X^T X \psi)}{N}
\]

Note: this is equivalent to the previous solution in terms of \( \phi \)

\[
\hat{\phi} = X \hat{\psi} = X (X^T X)^{-1} w
\]

\[
= (XX^T)^{-1} XX^T X (X^T X)^{-1} w
\]

\[
= (XX^T)^{-1} Xw,
\]
Bayesian case

\[ Pr(\psi) = \text{Norm}_\psi [0, \sigma_p^2 I] \]

\[ Pr(w | X, \theta) = \text{Norm}_w [X^T X \psi, \sigma^2 I] \]

Compute distribution over parameters:

\[
Pr(\psi | X, w, \sigma^2) = \frac{Pr(X | w, \psi, \sigma^2) Pr(\psi)}{Pr(X | w, \sigma^2)}
\]

Gives result:

\[
Pr(\psi | X, w, \sigma^2) = \text{Norm}_\psi \left[ \frac{1}{\sigma^2} A^{-1} X^T X w, A^{-1} \right]
\]

where

\[
A = \frac{1}{\sigma^2} X^T X X^T X + \frac{1}{\sigma_p^2} I
\]
Bayesian case

Predictive distribution:

\[ Pr(w^*|x^*, X, w) = \int Pr(w^*|x^*, \psi)Pr(\psi|X, w) \, d\psi \]

\[ = \text{Norm}_{w^*}\left[ \frac{1}{\sigma^2} x^T X A^{-1} X^T X w^*, x^T X A^{-1} X^T x^* + \sigma^2 \right] \]

where:

\[ A = \frac{1}{\sigma^2} X^T X X^T X + \frac{1}{\sigma_p^2} I \]

Notice that in both the maximum likelihood and Bayesian case depend on dot products \( X^T X \). Therefore this can be kernelized!
Relevance Vector Machine

Combines ideas of

- Dual regression (one parameter per training example)
- Sparsity (most of the parameters are zero)

Yields a model that only depends sparsely on training data.