# estimateBeta

Find beta coefficients to basis matrix H and the current set of hyperparameters theta as vector of s2f and sl, s2n represented by the current inverse of noisy covariance matrix Ky^-1 and the feature target vector y of the observations. It calculates several inverse Matrix products so instead passing the current Ky the function uses the infront decomposed lower triangle matrix L of Ky.

## Syntax

```
[beta, alpha0]= estimateBeta(H, L, y)
```

## Description

**[beta, alpha0]= estimateBeta(H, L, y)** compute polynom coefficients to solve mean approximation.

## Input Argurments

**H** basis matrix of training data. Polynomial represents of training data.

**L** lower triangle matrix of decomposed K matrix.

**y** regression targets.

## Output Argurments

**beta** beta coefficients for polynomial approximation with basis matrix**H**.

**alpha0** regression weights based on regression targets **y**.

## Requirements

- Other m-files required: None
- Subfunctions: chol, computeInverseMatrixProduct, computeTransposeInverseProduct
- MAT-files required: None

## See Also

- computeInverseMatrixProduct
- computeTransposeInverseProduct
- initKernelParameters



```
function [beta, alpha0]= estimateBeta(H, L, y)
    % Ky^-1 * y
    alpha0 = computeInverseMatrixProduct(L, y);

    % H * Ky^-1 * HT
    alpha1 = computeTransposeInverseProduct(L, H');

    % (H * Ky^-1 * HT)^-1 * H
    L1 = chol(alpha1, 'lower');
    alpha2 = computeInverseMatrixProduct(L1, H);

    % ((H * (Ky^-1 * HT))^-1 * H) * (Ky^-1 * y)
    beta = alpha2 * alpha0;
end
```

*Published with MATLAB® R2020b*