MACHINE LEARNING APPLIED TO SLV CALIBRATION

ADOPTING TECHNICS FROM MACHINE LEARNING

JÜRGEN HAKALA | LEONTEQ SECURITIES AG

MACHINE LEARNING APPLIED
TO SLV CALIBRATION

PROBLEM DEFINITION

LEVERAGE FUNCTION CALIBRATION IN SLV MODEL
DEFINITION

- Given a stochastic local volatility process

$$\begin{aligned} dS_t &= \mu(t)S_t dt + \sigma(S_t, t) f(V_t) S_t dW_t \\ dV_t &= \mu_V(V_t)dt + \xi\chi(V_t)dX_t \\ < dW_t, dX_t > &= \rho dt \end{aligned}$$

- As described in [GH] the calibration problem for the smile is to find a suitable leverage function that satisfies

$$\sigma^2_{Dupire}(S_t, t) = E^{P(S_t, V_t, \sigma)}(V_t | S = S_t)\sigma^2(S_t, t)$$

- Under the probability measure implied by the calibrated SLV process. Such problem is known as a McKean SDE.

SLV CALIBRATION
PROCEDURE

- Following [GH] the problem can be solved in a discretized MC setting. We use Euler discretization for demonstration purposes:

$$\begin{aligned}\Delta \ln(S_t) &= \mu(t)\Delta t - \frac{1}{2}\sigma^2(S_t, t)\Delta t \\ &\quad + \sigma(S_t, t) f(V_t)\left(\hat{\rho}\Delta W + \rho \Delta X\right) \\ \Delta V_t &= \mu_V(V_t)\Delta t + \xi\chi(V_t)\Delta X\end{aligned}$$

- Using the realization of MC up to t for N paths (or particles), we construct an approximation of the expectation in the calibration expression:

$$E^{P(S_t, V_t, \sigma)}(V_t | S = S_t) = R((S_t^1, V_t^1), \cdots, (S_t^N, V_t^N))(S)$$

SLV CALIBRATION
PROCEDURE II

- In [GH] the problem

$$E^{P(S_t,V_t,\sigma)}(V_t|S=S_t) = R((S_t^1,V_t^1),\cdots,(S_t^N,V_t^N))(S)$$

- was tackled by using kernel regression

$$R((S^1,V^1),\cdots,(S^N,V^N))(S) \quad = \quad \frac{\sum_{i=1}^N V_i K_h(S-S_i)}{\sum_{i=1}^N K_h(S-S_i)}$$

- In [vSGO] the estimation was tackled by binning and alternatively by regressing on a set of polynomials.

SLV CALIBRATION
CALIBRATION AS LEARNING

- This problem is a well known topic in machine learning and the proposed solution by [GH] is the standard method applied to such a problem.
- Nevertheless this method suffers from short-comings:
  - Bias in the areas close to the boundaries
  - Heavily depends on the choice of width parameter

- Explore alternative solutions than polynomials [vSGO] to the non-linear regression problem. Given independent samples of the realizations for the calibrated process (S_t, V_t) find a regression function for $E(V_t | S_t = S)$

MACHINE LEARNING APPLIED
TO SLV CALIBRATION

MACHINE LEARNING

SLV CALIBRATION
CALIBRATION AS LEARNING

- The core problem of estimating a function based on examples is a well studied one.
- For examples without noise the problem can be reduced to interpolation. It is an ill-posed problem which can be made unique by defining a regularizer

$$\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda||Pf||^2$$

- G is the solution to the Green's function of the operator P'P.

$$\hat{P}PG(x,\xi) = \delta(x-\xi) \qquad \text{and} \qquad f(x) = \sum_{i=1}^{N} c_i G(x, x_i)$$

- And coefficients c_i are the solution of the normal equation.

$$(G + \lambda 1)c = y$$

MACHINE LEARNING

MACHINE LEARNING – SUPERVISED LEARNING

- Function approximation and regression is a subset of machine learning problems and associated methods

- In ML terms this is called supervised learning
- Samples are presented to the algorithm to “learn” the underlying relationship. Usually the set of available samples is split into training, validation and test set.
- The validation set is used to determine certain (meta)-parameters of the training method.
- The (independent) test set is used to determine the performance of the algorithm

MACHINE LEARNING
SUPERVISED LEARNING – NON-PARAMETRIC

- So called non parametric methods do not assume a specific parametric form of the function to be approximated. Kernel regression is the most prominent of these methods.

$$R((x_1, y_1), \cdots, (x_N, y_N))(X) \quad = \quad \frac{\sum_{i=1}^{N} y_i K_h(x - x_i)}{\sum_{i=1}^{N} K_h(x - x_i)}$$

- The kernel is semi-positive and is constrained to satisfy

$$\begin{aligned} K(x) &\geq 0 \\ K(x) &= K(-x) \\ \int_{-\infty}^{\infty} K(x)dx &= 1 \\ K_h(x) &= \frac{1}{h} K\left(\frac{x}{h}\right) \end{aligned}$$

MACHINE LEARNING

SUPERVISED LEARNING – KERNELVARIANTS

Kernel	Formula	Range
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	unlimited
Quartic	$\frac{15}{16}\left(1-x^2\right)^2$	$\lvert x\rvert \leq 1$
Epanechnikov	$\frac{3}{4}\left(1-x^2\right)$	$\lvert x\rvert \leq 1$
Sigmoid	$\frac{2}{\pi}\frac{1}{e^u+e^{-u}}$	unlimited

MACHINE LEARNING
SUPERVISED LEARNING – NON-PARAMETRIC

- Local Linear Kernel regression is the second most prominent of these methods but requires slightly more computation.

$$R((x_1, y_1), \cdots, (x_N, y_N))(x) \quad = \quad \min_{\alpha,\beta} \sum_{i=1}^{N} (y_i - \alpha - x_i\beta)^2 K_h(x - x_i)$$

- The LMS solution is found by summing 4 terms over all samples and solving a 2x2 linear system.

MACHINE LEARNING
SUPERVISED LEARNING – NON-PARAMETRIC II

- The Nadarayan-Watson based Kernel regression suffers from some shortcomings
- All “examples” are used, there is no compression
- At the boundaries there is a systematic bias
- Alternatives:
  - Linear Kernel Regression – suffers much less from bias at the boundaries
  - Parametric – tricky to guess a good general parametric form
- Largest issue is the choice of bandwidth
  - Silverman’s rule of thumb
  - Cross validation – in particular leave one out cross validation

MACHINE LEARNING

SUPERVISED LEARNING – MODEL SELECTION

Largest issue is the choice of bandwidth

- Silverman's rule of thumb

$$h = \left(\frac{4\sigma^5}{3n} \right)^{\frac{1}{5}}$$

- Cross validation – esp. leave one out cross validation

$$h_{opt} = \min_h \left(\sum_{i=1}^{N} \left(y_i - K_h{}^{(-i)}(x_i) \right)^2 \right)$$

- The simple Silverman rule of thumb often leads to suboptimal results and cross validation is pretty expensive computationally.

MACHINE LEARNING

SUPERVISED LEARNING – FUNCTION APPROXIMATION

There is a large variety of approaches to estimate functions from examples.

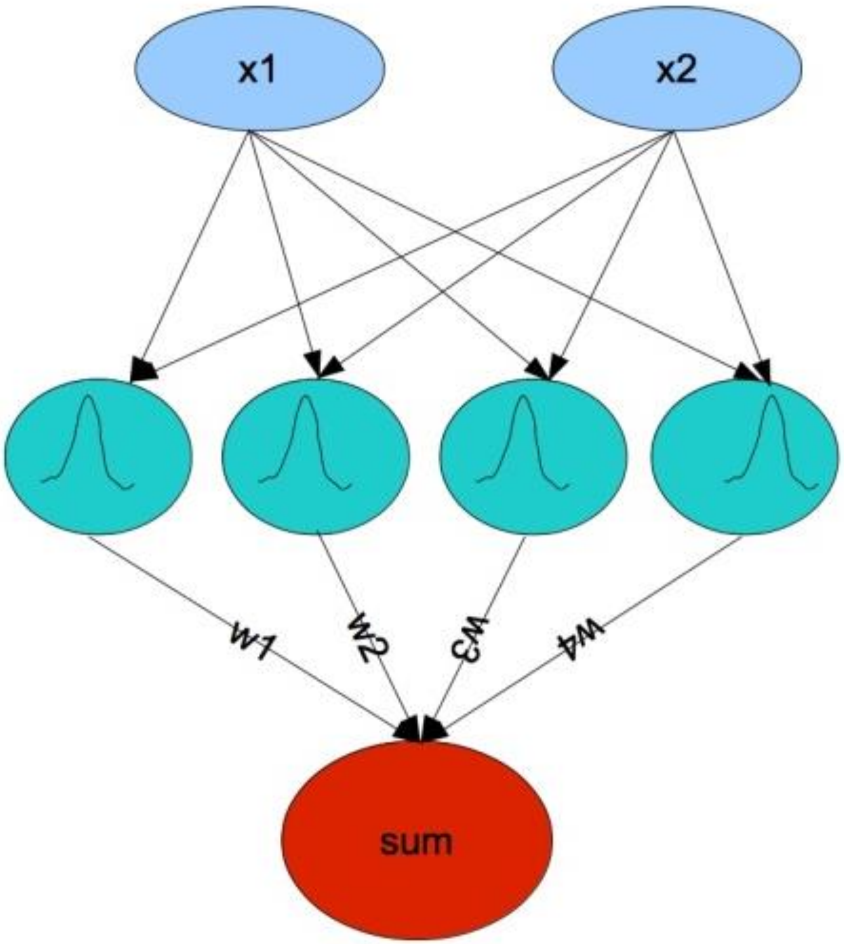
- Radial Basis Functions and partition of unity RBF

$$
\begin{aligned}
RBF(x) &= \sum_{i=1}^{C} w_i K_{h_i}(x - c_i) \\
PURBF(x) &= \frac{\sum_{i=1}^{C} w_i K_{h_i}(x - c_i)}{\sum_{i=1}^{C} K_{h_i}(x - c_i)}
\end{aligned}
$$

- Radial Basis Functions are quite close to Kernel regression as the functions used are of the same type. But it reduces the computational burden by taking a small number of kernels compared to the number of examples.
- Training or determination is needed for the placement of centers and determination of the width as well as the weights.

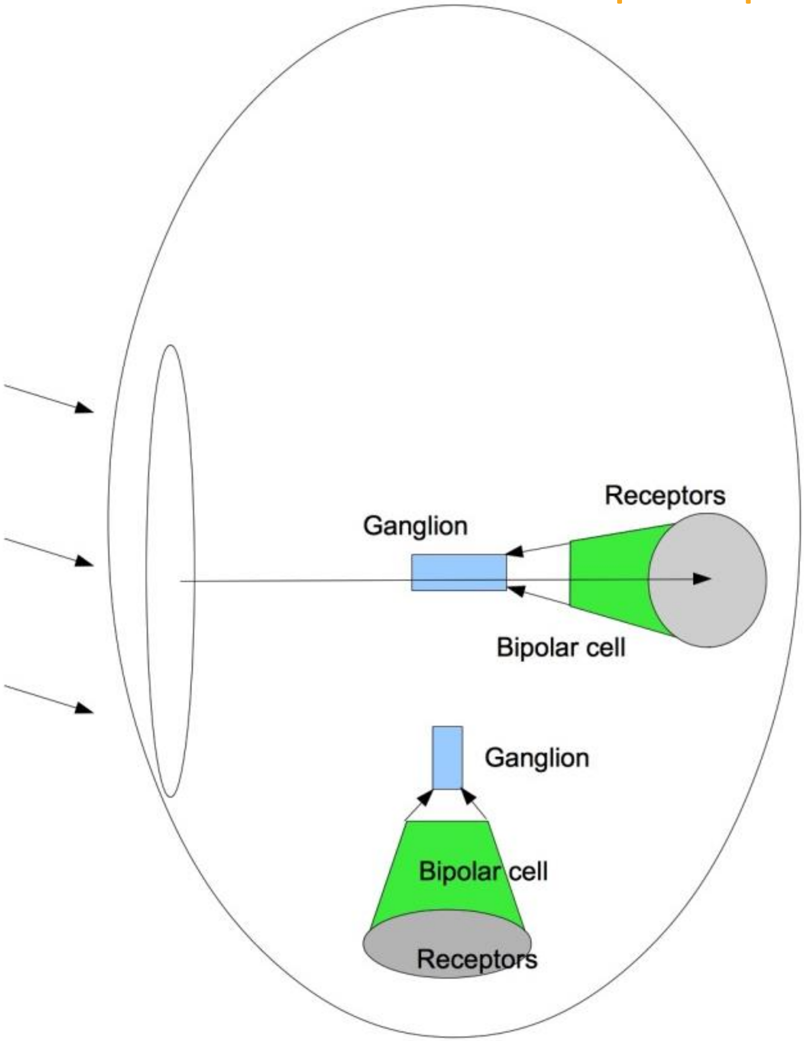
MACHINE LEARNING

SUPERVISED LEARNING | RBF | MOTIVATION FROM BIOLOGY



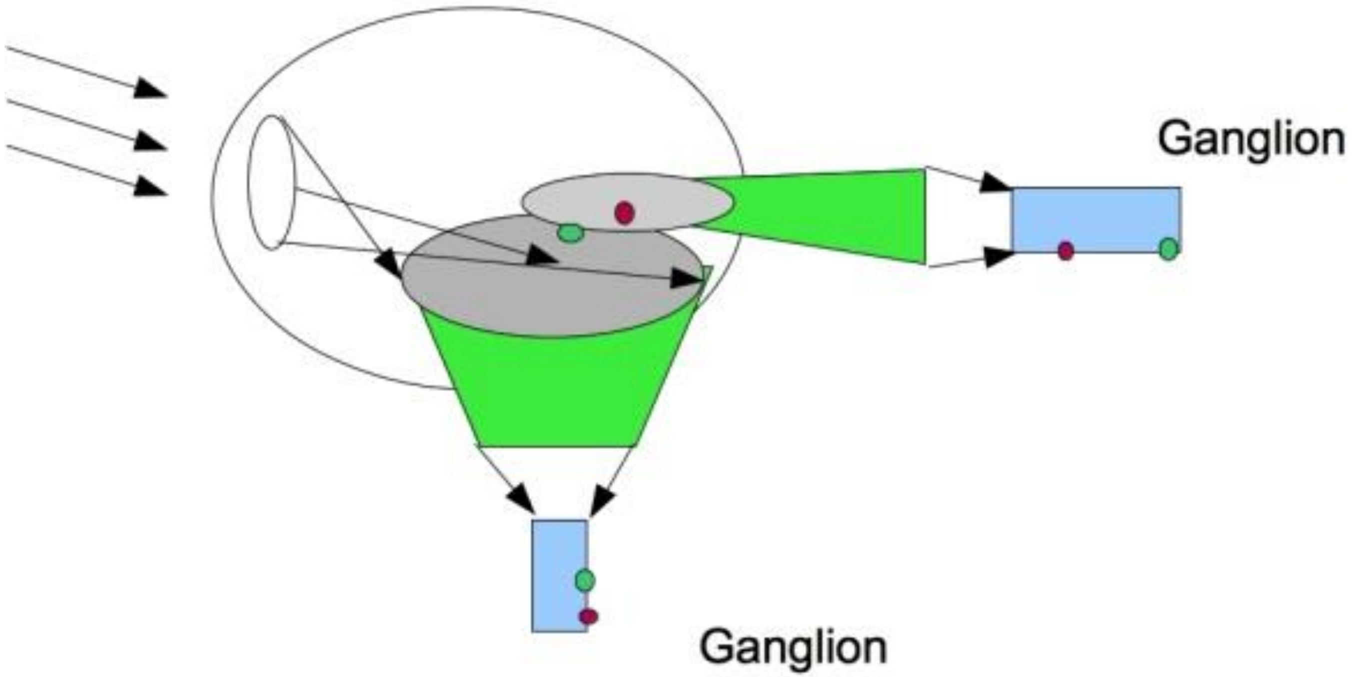
MACHINE LEARNING

SUPERVISED LEARNING | RBF | MOTIVATION FROM BIOLOGY



MACHINE LEARNING

SUPERVISED LEARNING | RBF | MOTIVATION FROM BIOLOGY



MACHINE LEARNING
SUPERVISED LEARNING – RBF - TRAINING

Training of weights for RBF

- Least square problem

$$LS = \frac{1}{2N}\sum_{i=1}^{N}(y_i - RBF(x_i))^2$$

- Leads to normal equation but with size C

$$w_i = (A_{ij}^T A_{ij})^{-1} b_j$$

- With

$$\begin{aligned} A_{ij} &= K_{h_j}(x_i - c_j) \\ b_j &= \sum_{i}^{N} y_i K_{h_j}(x_i - c_j) \end{aligned} \quad (1)$$

- Alternative would be stochastic gradient descent if the training data cannot be presented as a whole.

MACHINE LEARNING
SUPERVISED LEARNING – RBF – TRAINING

- Often the matrix will be badly conditioned, hence a normalizer is a prudent choice

$$LSN = \frac{1}{2N}\sum_{i=1}^{N}(y_i - RBF(x_i))^2 + \lambda\sum_{j=1}^{C} w_j^2$$

$$w_i = (A_{ij}^T A_{ij} - \lambda id)^{-1} b_j$$

- The regularizer can be found by cross validation.

MACHINE LEARNING

SUPERVISED LEARNING – RBF - CENTERS

Determination of centers for RBF

- Select the centers as a subset of the training examples (plus min and max)
- Stochastic gradient descent
- Resource allocation – gradually enlarge the number of basis functions to allocate more densely in areas which are difficult to fit

MACHINE LEARNING

SUPERVISED LEARNING – RBF – WIDTH

Determination of width for RBF

- Select the width as the average distance to the k-nearest neighbors
- Select a global width (difficult to cross validate, split into a training and validation set – losing examples)

MACHINE LEARNING

SUPERVISED LEARNING – RBF - PRUNING

- To avoid overfitting and bad conditioning of the regression problem pruning can be used.
- Kernels with centers too close to each other will be merge, pruning degrees of freedom from the approximator.
- Pruning candidate criterion: $\min_i \left(\frac{|c_i - c_j|}{h_j} \right) \leq \Theta$
- If neighboring centers are pruning candidates just prune one of them

MACHINE LEARNING

SUPERVISED LEARNING – ALTERNATIVES?

Alternative could be multi-layer perceptron / Deep Networks

- Training is much more demanding, multiple epochs of stochastic gradient based training.
- Model selection is quite tricky – number of layers, number of units in each layer, weight sharing, activation functions,….
- Consider MLP type learning machine too demanding for this rather limited application.

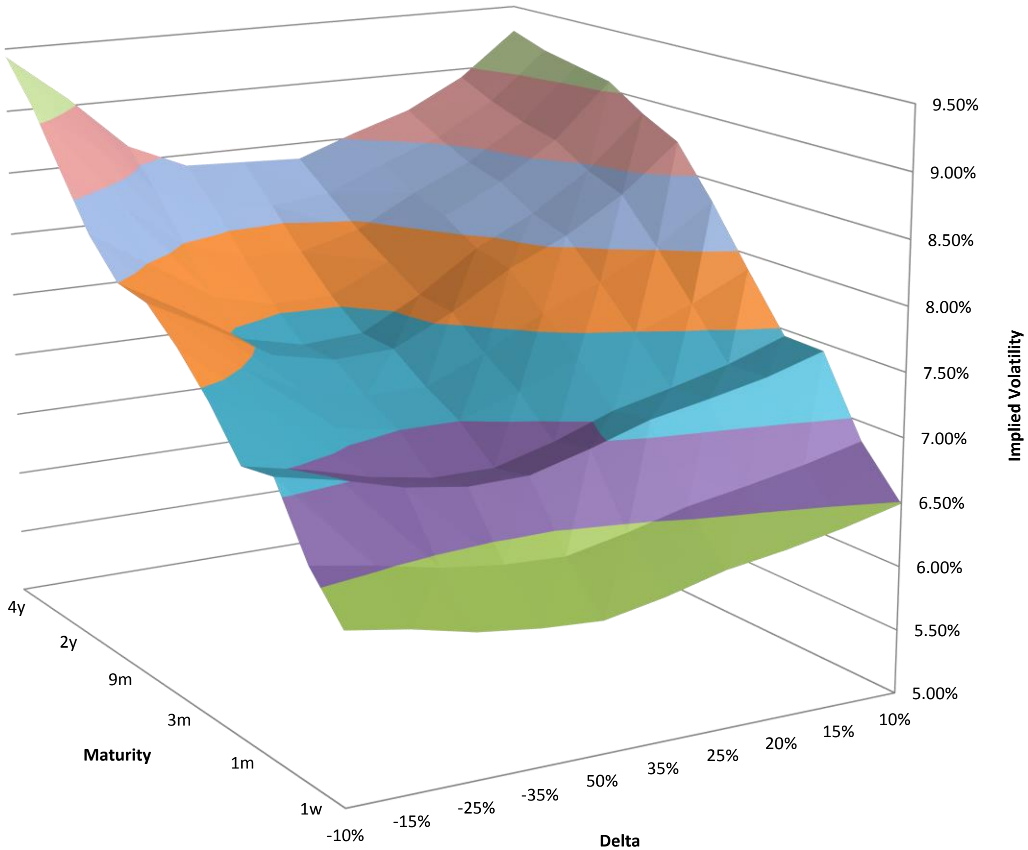
MACHINE LEARNING
COMPUTATIONAL EFFORT

- For standard Kernel Regression it is mainly due to sorting O(n log(n)) [GH], then the lookup can be optimized. Optimal determination of width (cross-validation) requires the evolution of all kernels at all points several times – very costly.
- Local Linear requires sorting and inversion of a matrix.
- RBF – PU-RBF
  - Training requires the solution of a small linear system
  - Sorted samples can be used to optimize the training (matrix and rhs are sums over samples weighted by kernel)
  - Width and pruning determination requires local computation of the order #kernels

EXAMPLES

EXAMPLES
EUR/USD

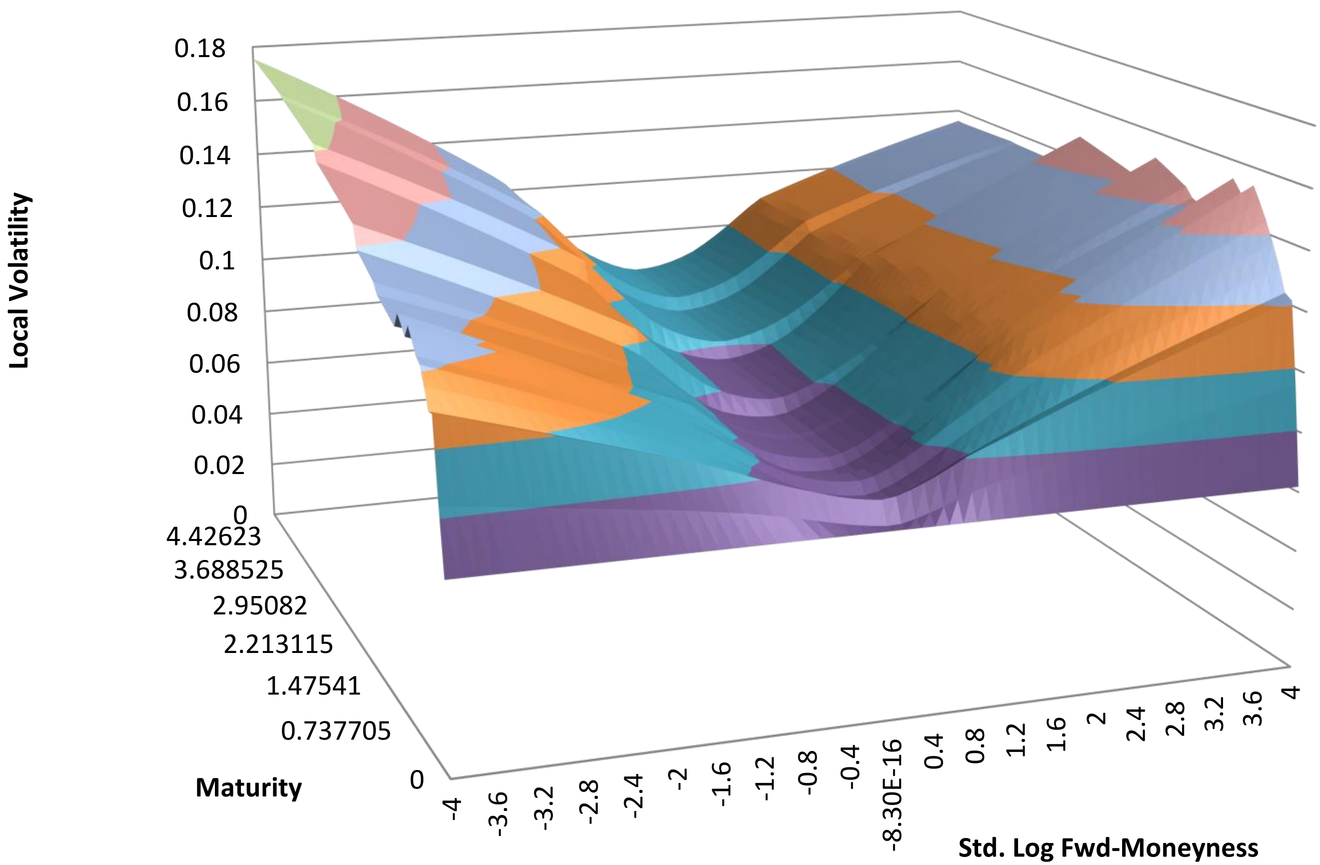
Market data: EUR/USD Volatility



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD

Local Volatility: EUR/USD Local Volatility

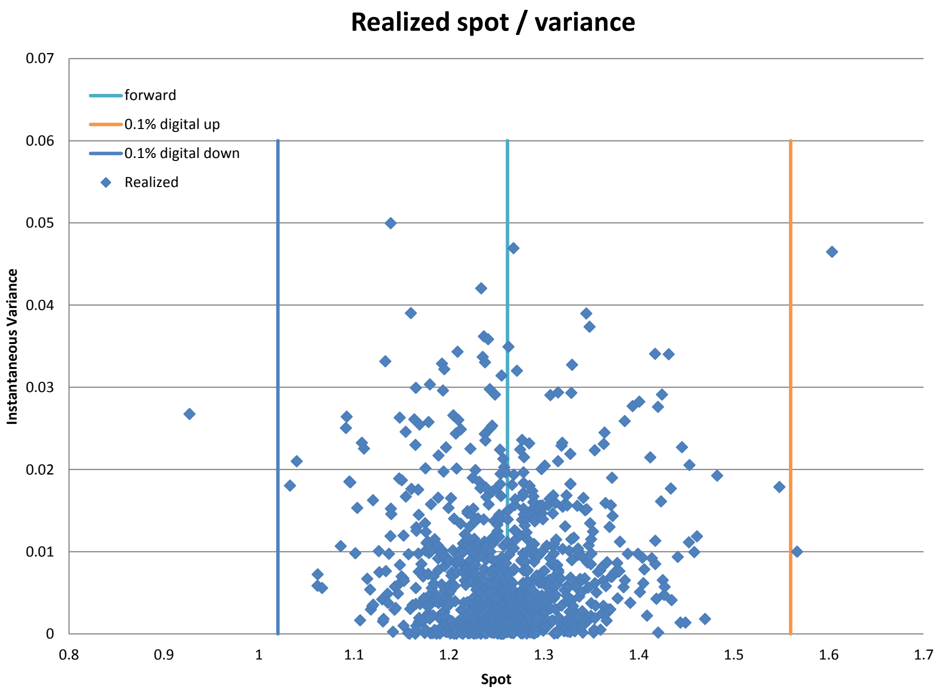


Source: Leonteq AG, internal data, 27.03.2018

MACHINE LEARNING APPLIED TO SLV CALIBRATION

EXAMPLES
EUR/USD 6M

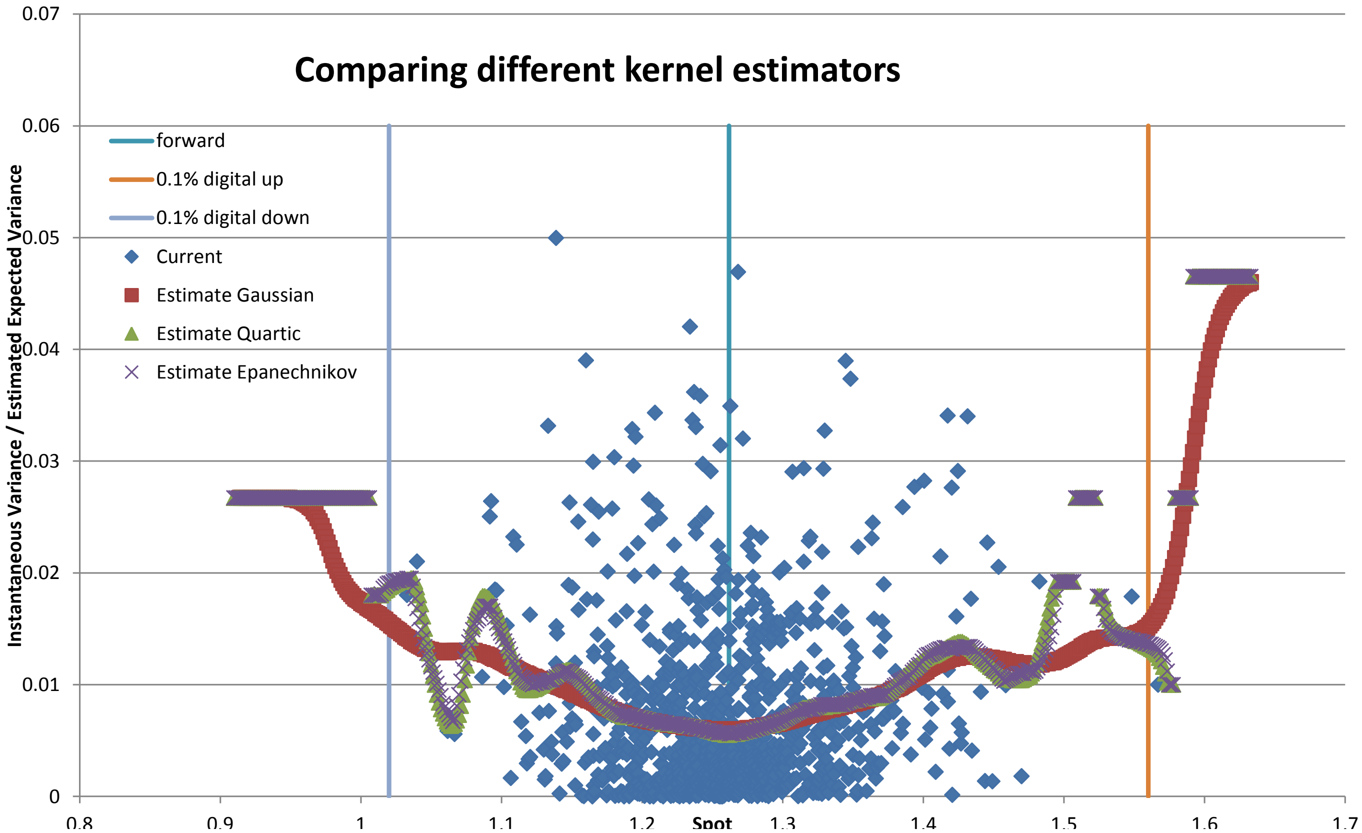
- Calibrate the Heston SV model on 6M maturity, mixing weight 90%.
- Use the particle method with standard settings: 1024 particles, digital bound 0.1%, Kernel width determined by Silverman's 'Rule of Thumb', Gaussian Kernels



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD 6M

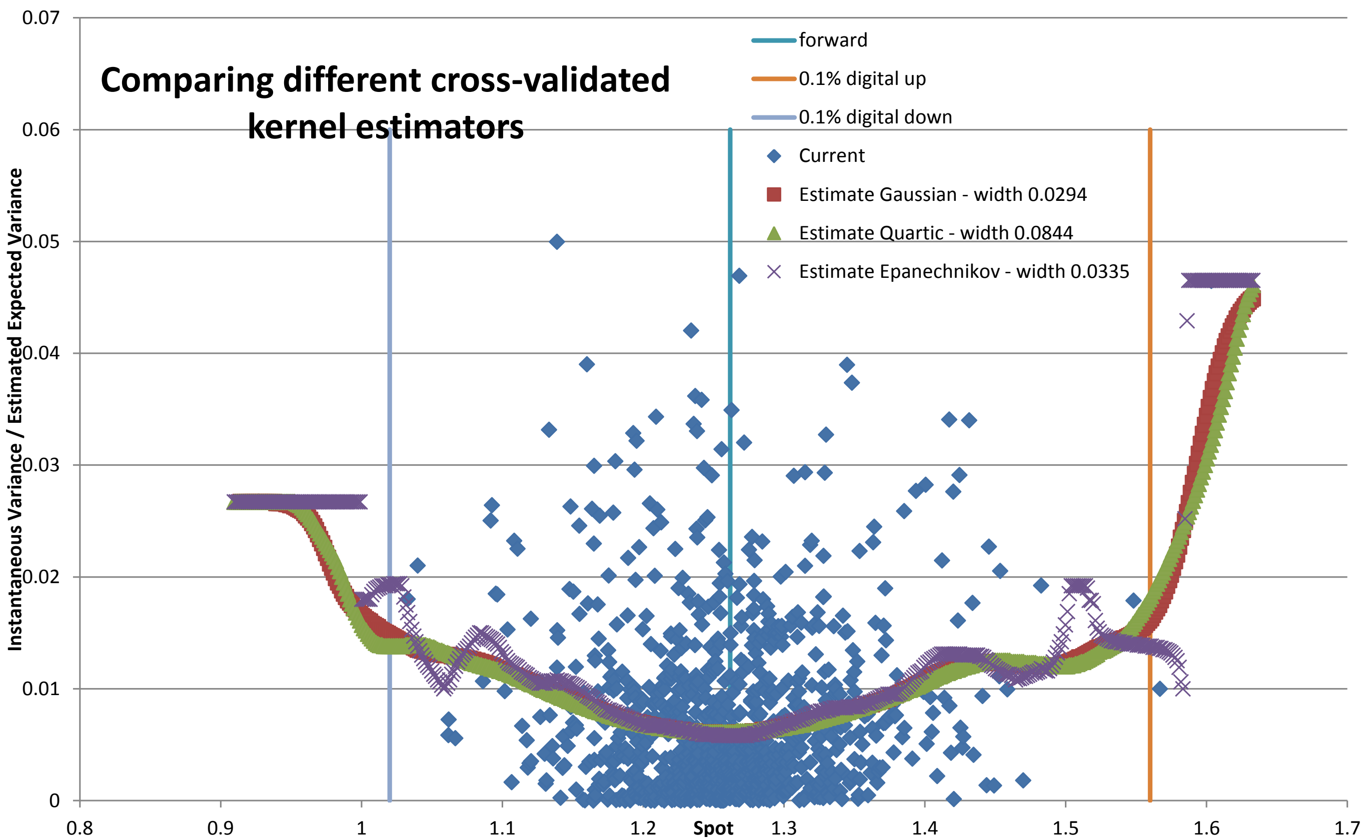
Compare Kernel Estimators



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD 6M

Compare Kernel Estimators with Cross Validated Width

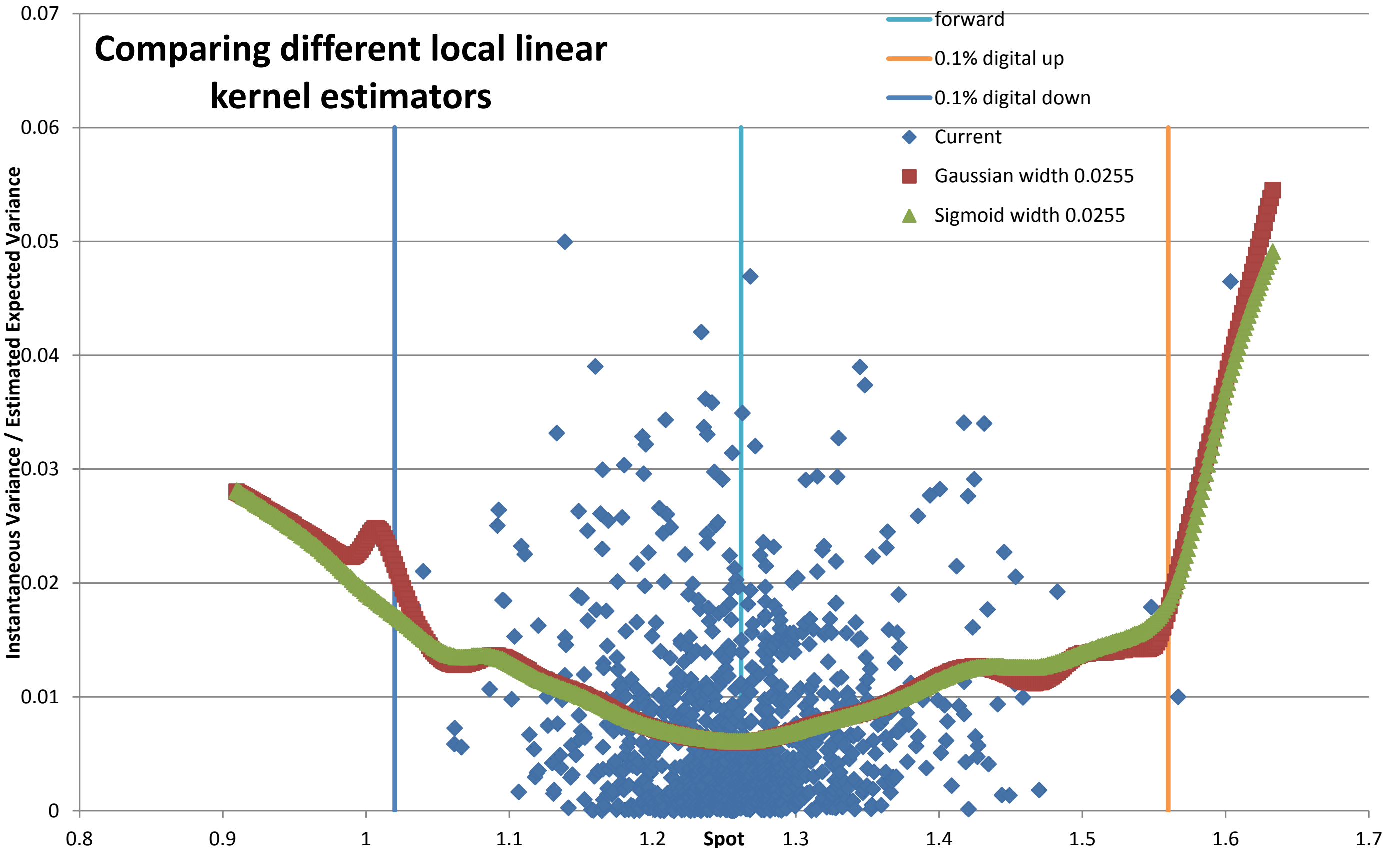


Source: Leonteq AG, internal data, 27.03.2018

MACHINE LEARNING APPLIED TO SLV CALIBRATION

EXAMPLES
EUR/USD 6M

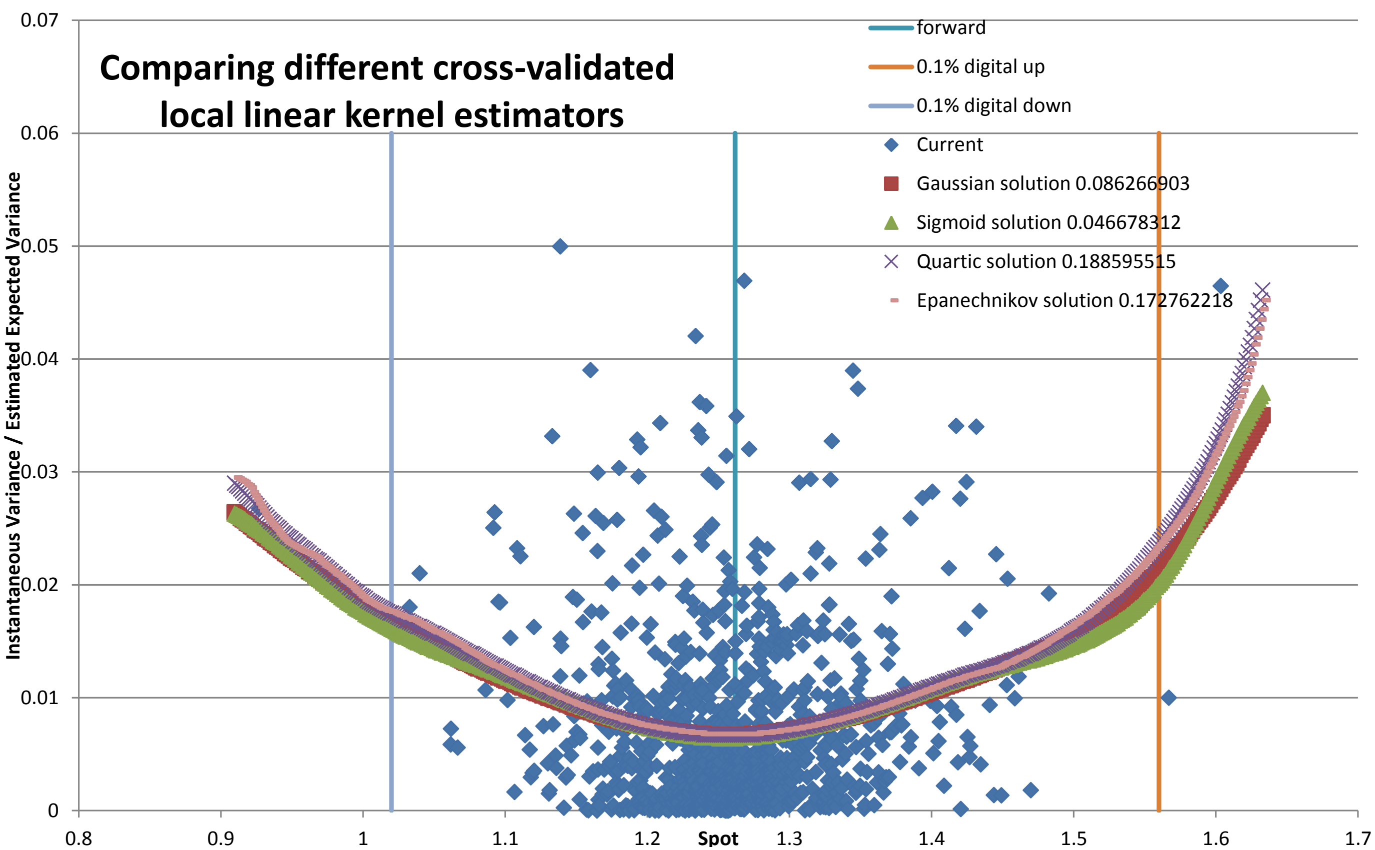
Compare Local Linear Kernel Estimators



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD 6M

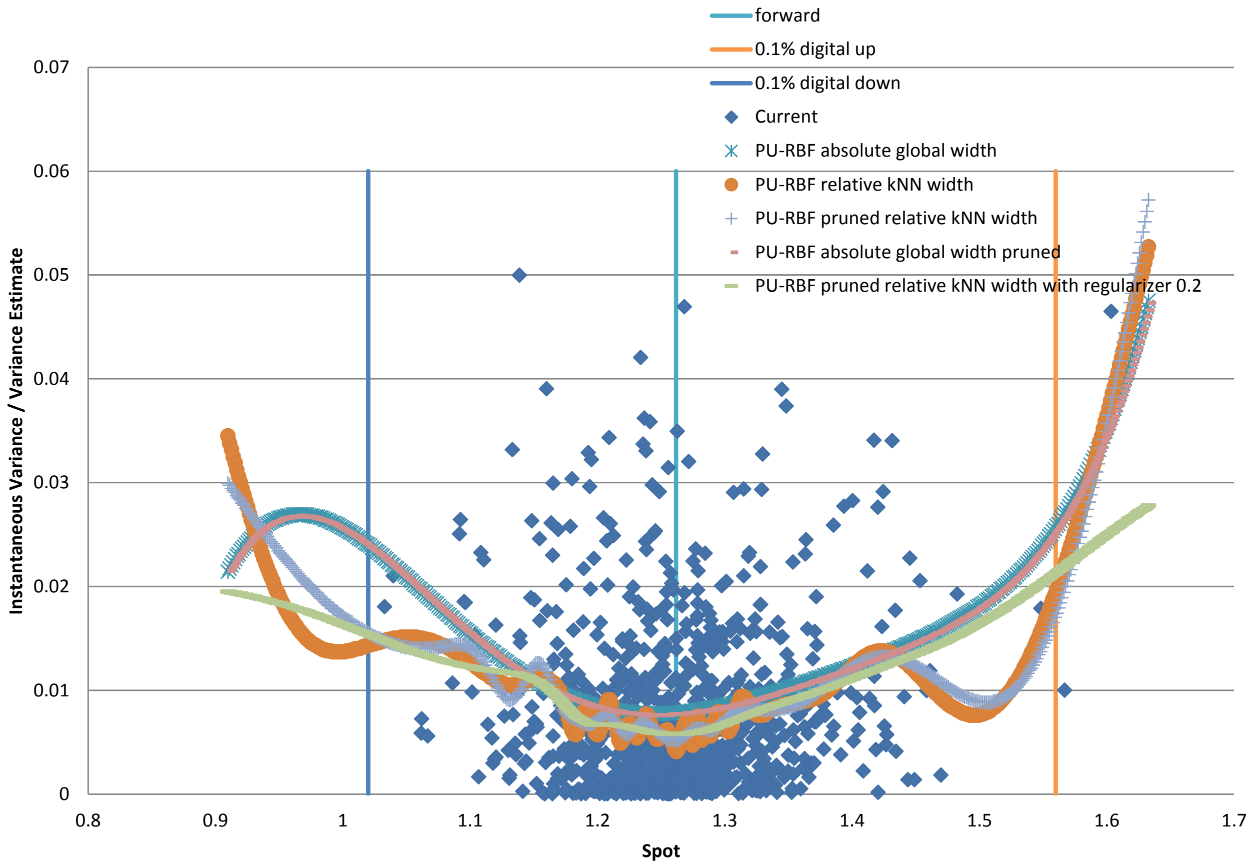
Compare Local Linear Kernel Estimators with Cross Validated Width



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD 6M

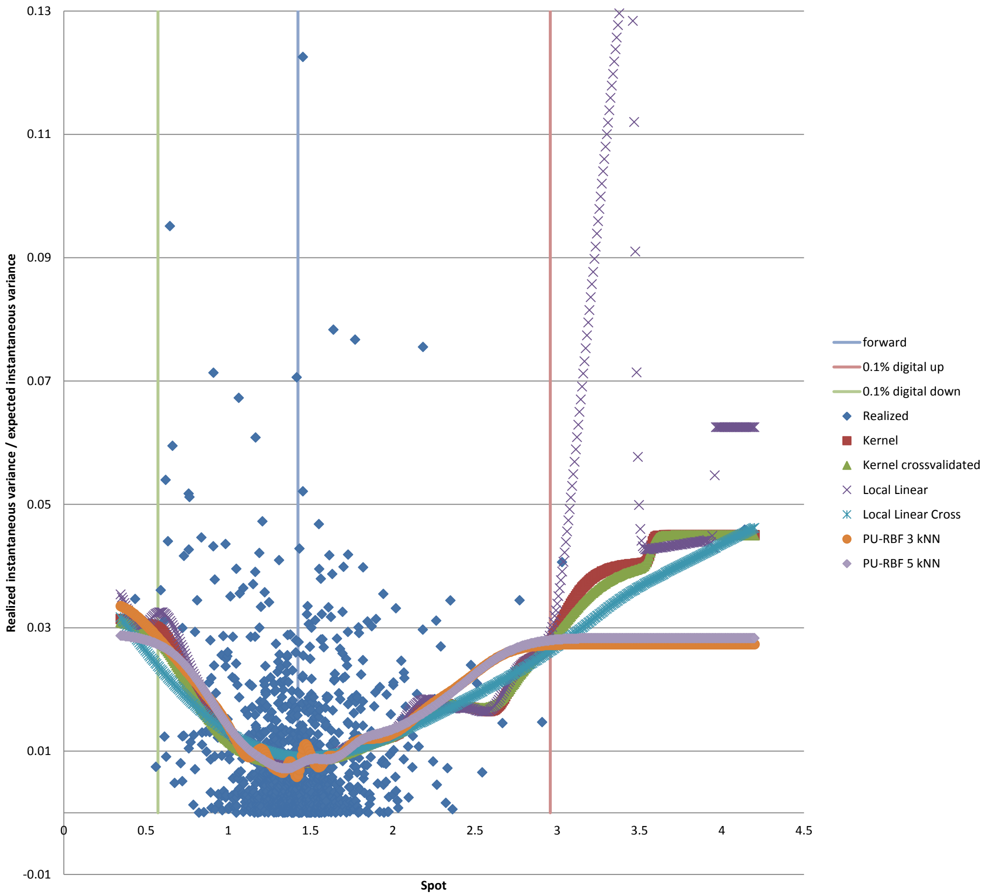
Gaussian PU-RBF



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/USD 5Y

Compare Kernel vs PU-RBF: Estimation of expected variance

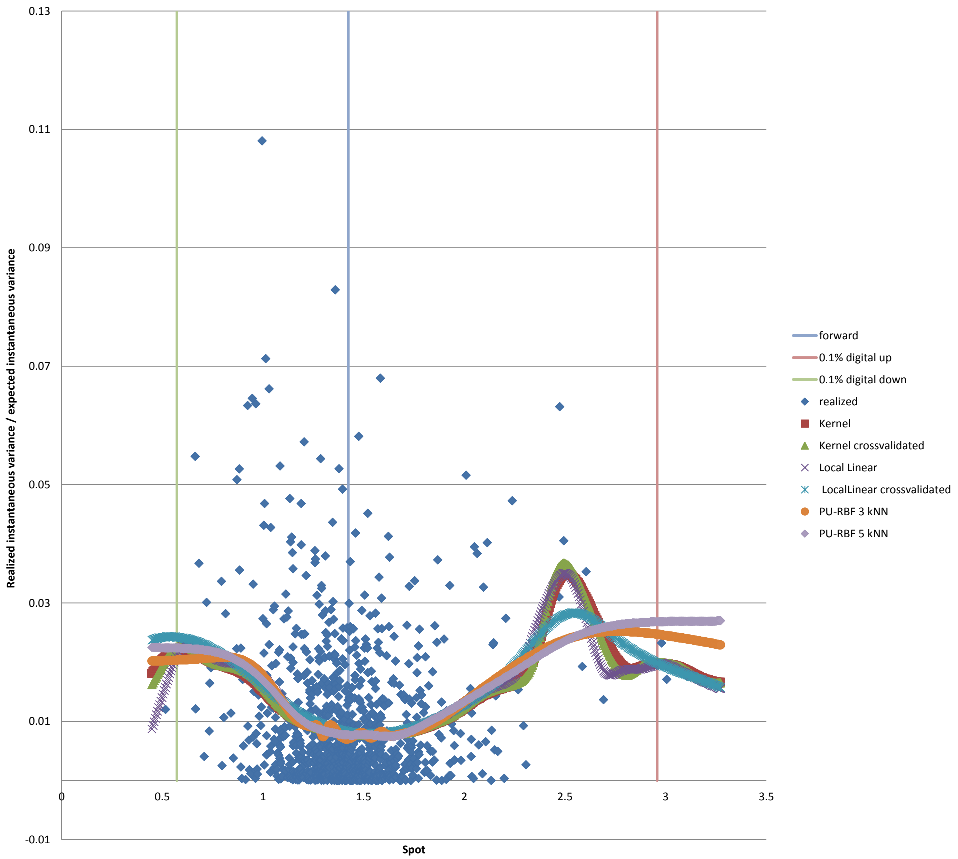


Source: Leonteq AG, internal data, 27.03.2018

MACHINE LEARNING APPLIED
TO SLV CALIBRATION

EXAMPLES
EUR/USD 5Y

Compare Kernel vs PU-RBF: Estimation of expected variance

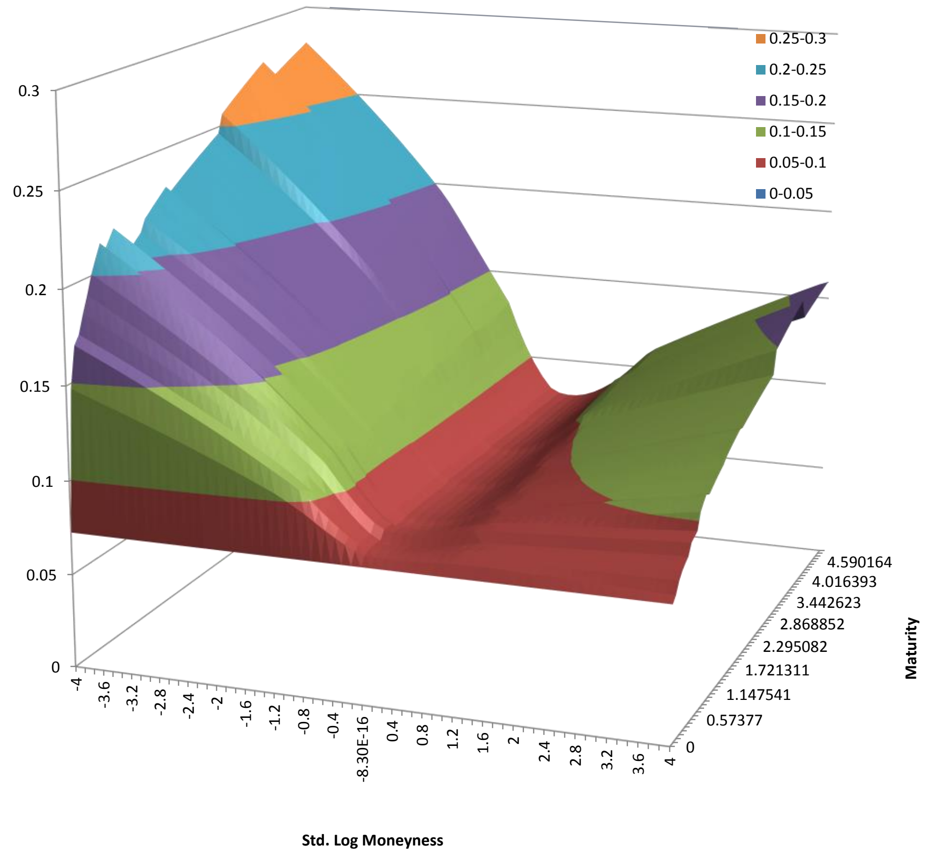


Source: Leonteq AG, internal data, 27.03.2018

MACHINE LEARNING APPLIED
TO SLV CALIBRATION

EXAMPLES
USD/JPY

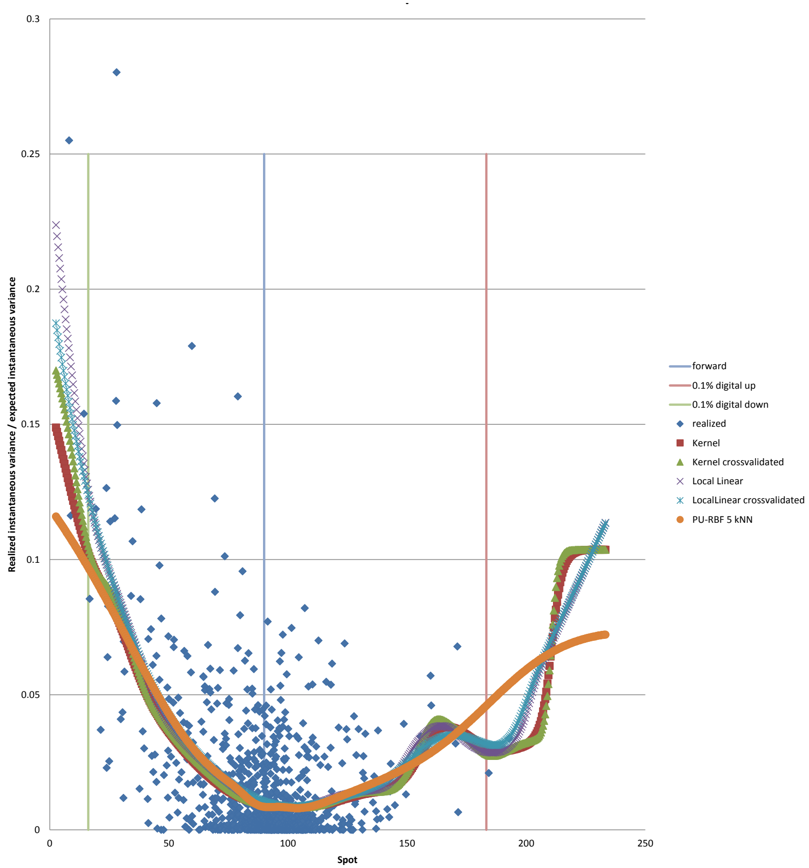
USD/JPY Local volatility



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
USD/JPY 5Y

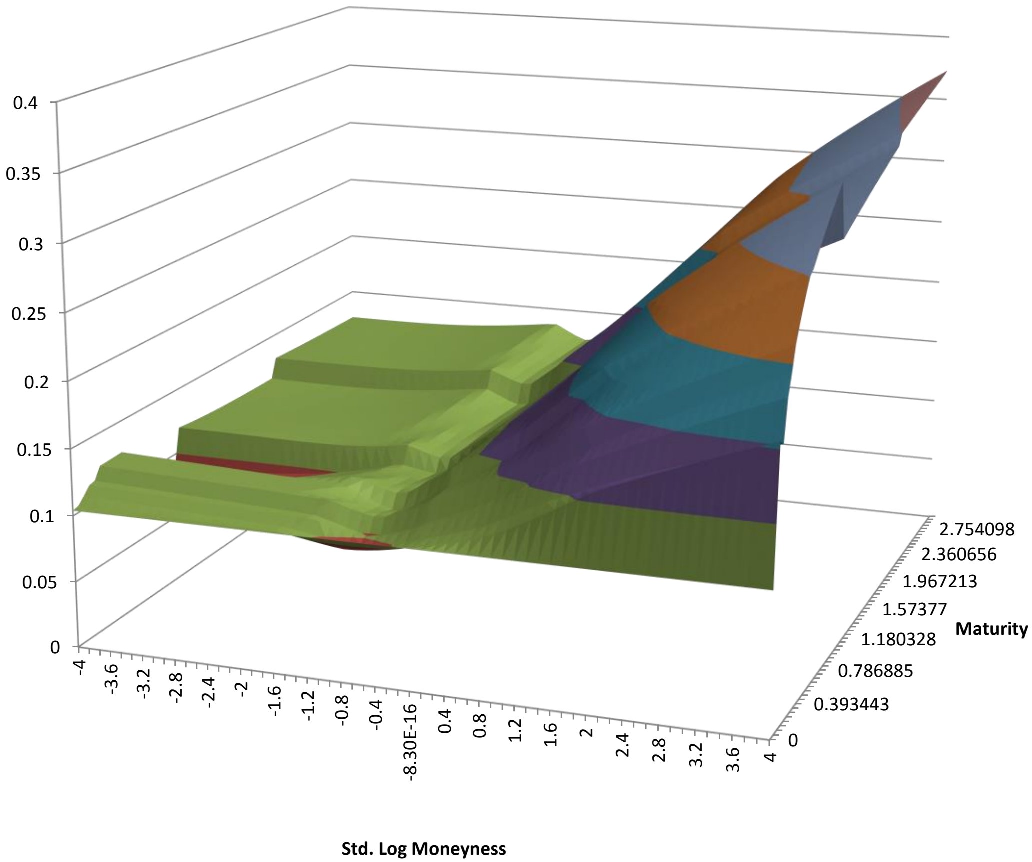
Compare Kernel vs PU-RBF: Estimation of expected variance



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/BRL

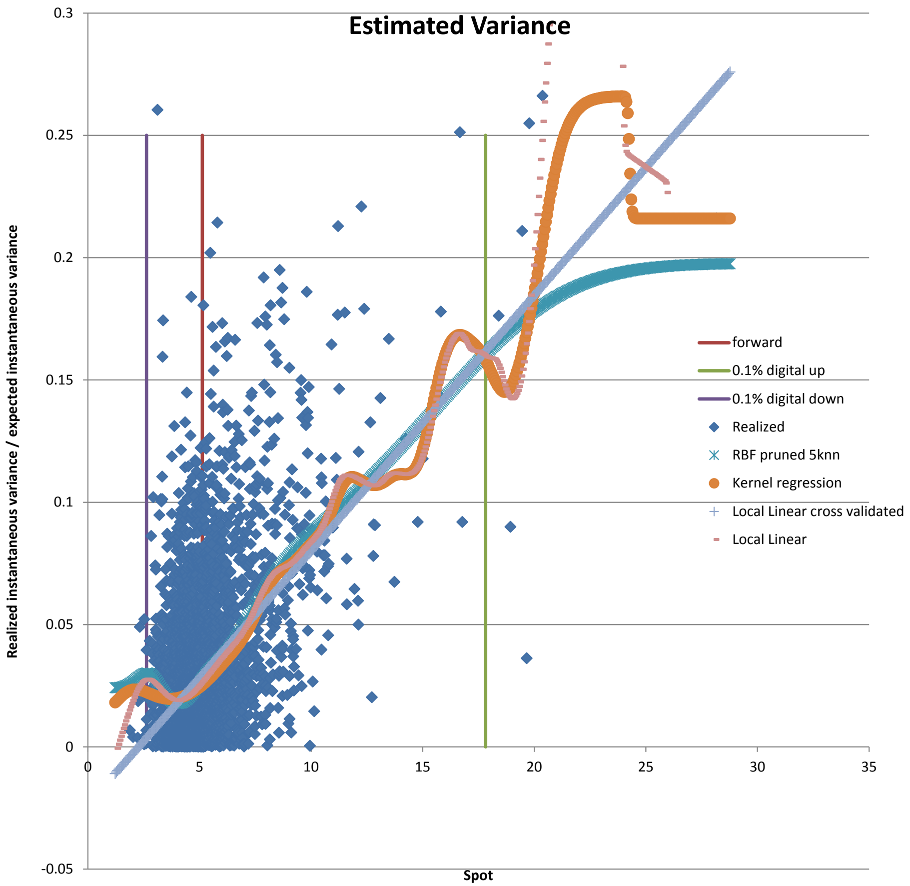
Local volatility: EUR/BRL Local Volatility



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
EUR/BRL 3Y

Compare Kernel vs PU-RBF: Estimated variance



Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES
CALIBRATION

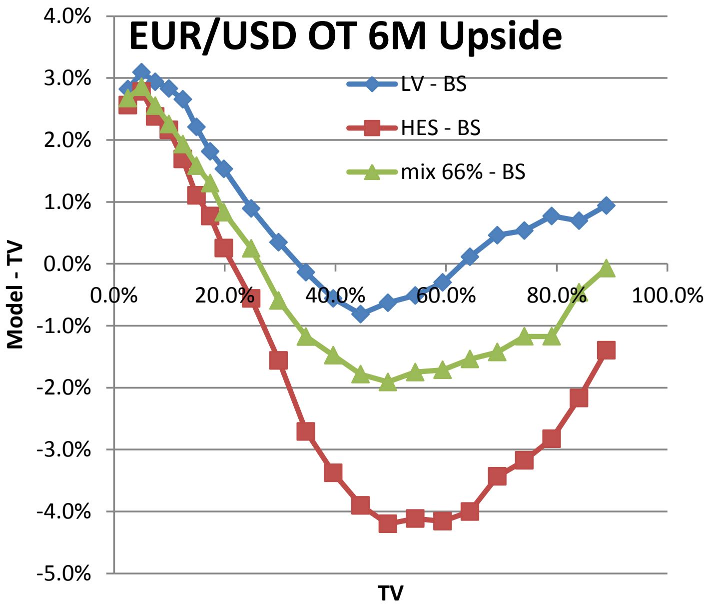
As the best results were obtained with kNN = 5, pruning $\Theta = 0.4$, regularizer $\lambda = 0.2$ and relative width 1.6 we will use those.

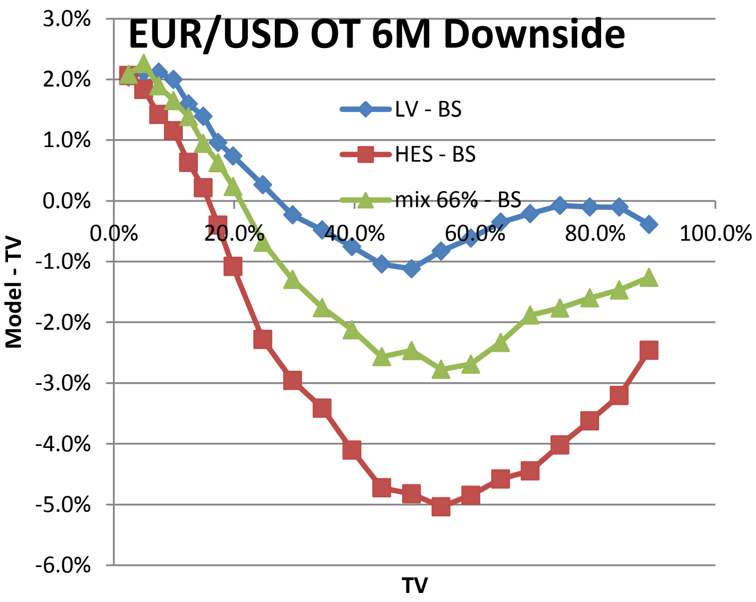
- The calibration error is of the order of some basis points.
- Compared to Kernel Regression the RMS error is smaller, often by a factor of 2

Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES –EXOTICS PRICING

EUR/USD ONE-TOUCH

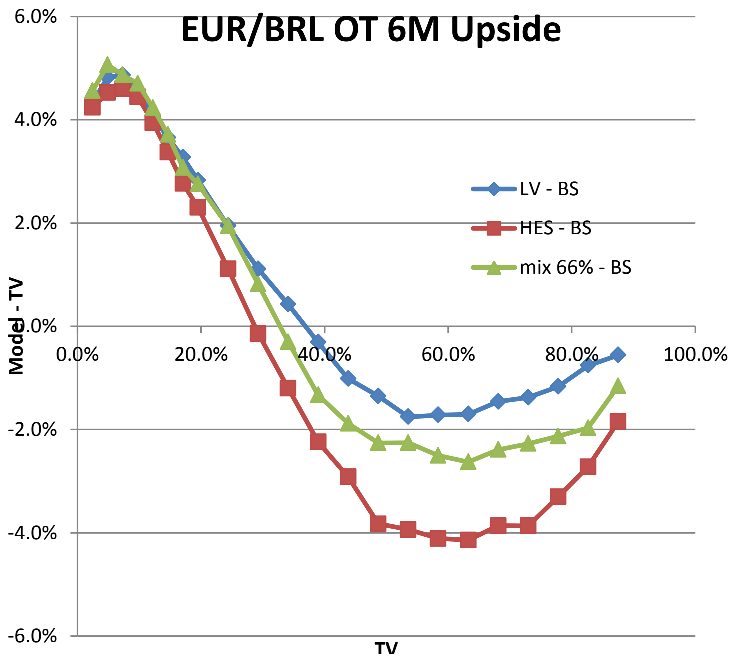


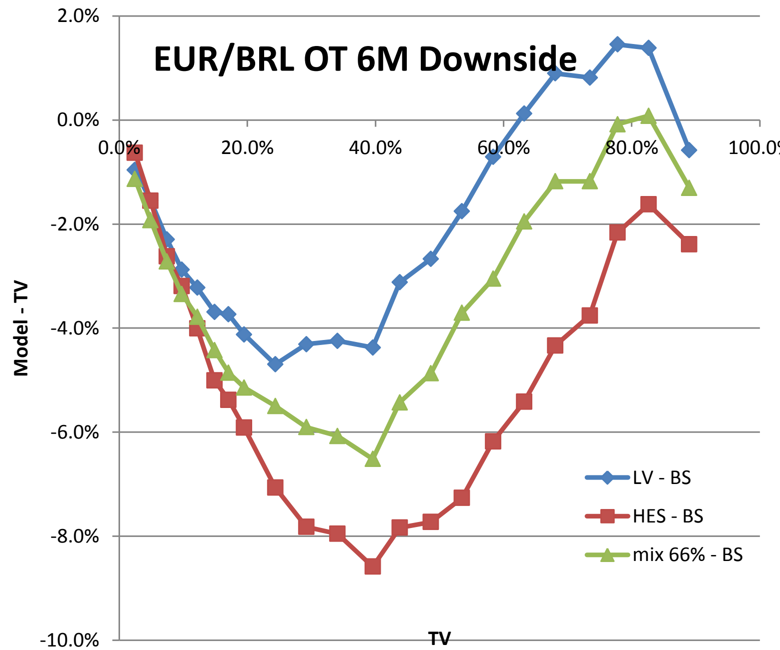


Source: Leonteq AG, internal data, 27.03.2018

EXAMPLES –EXOTICS PRICING

EUR/BRL ONE-TOUCH





Source: Leonteq AG, internal data, 27.03.2018

MACHINE LEARNING APPLIED
TO SLV CALIBRATION

CONCLUSION

CONCLUSION

Machine Learning methods from the supervised learning field can be employed to approximate the conditional expected variance required in the SLV calibration.

The advantage of PU-RBF is a robust smooth approximation that «automatically» adapts to the input density with a restricted set of basis functions.

The training effort is relatively small, requiring the inversion of a rather small matrix. The required number of samples (particles) to train RBF networks is smaller than for the Kernel Regression case.

No a-priori selection of a set of polynomials is necessary, as such the method is less susceptive to a prior bias.

OUTLOOK

We will be further accelerating the method by using better sorting algorithms for the samples, in particular as the samples are already pre sorted from the last step.
Larger step-size for the time discretization would be desirable and will be a future topic in our refinements.

BIBLIOGRAPHY

[GH] Guyon, Julien and Henry-Labordere, Pierre, The Smile Calibration Problem Solved (July 13, 2011). Available at SSRN: https://ssrn.com/abstract=1885032or http://dx.doi.org/10.2139/ssrn.1885032

[PG] Poggio T., Girosi F.Networks for approximation and learning. Proc. IEEE., 78 (9) (1990), pp. 1481-1497.

[vSGO] van der Stoep, Anthonie and Grzelak, Lech A. and Oosterlee, Cornelis W., The Heston Stochastic-Local Volatility Model: Efficient Monte Carlo Simulation (November 19, 2013). International Journal of Theoretical and Applied Finance, Vol. 17, No. 7 (2014).. Available at SSRN: https://ssrn.com/abstract=2278122 or http://dx.doi.org/10.2139/ssrn.2278122

THANK YOU

LEGAL DISCLAIMER

This publication serves only for information purposes and is not research; it constitutes neither a recommendation for the purchase of financial instruments nor an offer or an invitation for an offer. No responsibility is taken for the correctness of this information. Investors bear the full credit risk of the issuer/guarantor for products which are not issued as COSI® products. Before investing in derivative instruments, investors are highly recommended to ask their financial advisor for advice specifically focused on the investor´s financial situation; the information contained in this document does not substitute such advice.

This publication does not constitute a simplified prospectus pursuant to art. 5 CISA, or a listing prospectus pursuant to art. 652a or 1156 of the Swiss Code of Obligations. The relevant product documentation can be obtained directly at Leonteq Securities AG via telephone +41 (0)58 800 1111, fax +41 (0)58 800 1010, or via e-mail: termsheet@leonteq.com.

Selling restrictions apply for the **EEA**, **Hong Kong**, **Singapore**, the **USA**, **US persons**, and the **United Kingdom** (the issuance is subject to Swiss law).
The Underlying's performance in the past does not constitute a guarantee for their future performance. The financial products' value is subject to market fluctuation, which can lead to a partial or total loss of the invested capital. The purchase of the financial products triggers costs and fees. Leonteq Securities AG and/or another related company may operate as market maker for the financial products, may trade as principal, and may conclude hedging transactions. Such activity may influence the market price, the price movement, or the liquidity of the financial products.

Any - including only partial - reproduction of any article or picture is solely permitted based on an authorization from Leonteq Securities AG. No responsibility is assumed in case of unsolicited delivery.

LEONTEQ SECURITIES AG
Europaallee 39 | CH-8004 Zurich | info@leonteq.com | www.leonteq.com