Tuning the hyperparameters in an MR patch-based CT metal artifact reduction algorithm

Jonathan Scharff Nielsen[1,2], Jens Morgenthaler Edmund[2,3] and Koen Van Leemput[1,4]

(1): Image Analysis and Computer Graphics Section, DTU Compute, Lyngby; (2): Radiotherapy Research Unit, Department of Oncology, Herlev; (3) Niels Bohr Institute, University of Copenhagen, Copenhagen; (4) Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Boston MA.

josni@dtu.dk

Introduction

Metal implants cause streak and cupping artifacts on Computed Tomography (CT) images that may be addressed using metal artifact reduction algorithms (MARs). For this purpose may be used the superior anatomical information of a coregistered Magnetic Resonance image (MR), which has led to the development of the kerMAR (kernel regresion MAR) algorithm. This algorithm estimates uncorrupted CT values by combining kernel regression on MR patch / CT value pairs in the uncorrupted patient volume with a forward model of the CT artifacts.

In this work we present the generative model behind the kerMAR algorithm and consider one of the optimisation problems important to using it: The estimation of the model hyperparameters by Empirical Bayes estimation and the Expectation Maximisation (EM) algorithm[1]. Such data-driven estimation of the hyperparameters ensures self-containment of the algorithm and allows for its easy incorporation in other MAR algorithms that rely on Bayesian principles such as Maximum A Posteriori (MAP) estimation.

The kerMAR model

Consider a set of voxels with indices $i \in \mathcal{T}$ in a patient volume for which a CT and MR have been acquired. For these voxels we consider the set $\{t_i, \mathbf{m}_i\}_{i\in\mathcal{T}}$ of measured CT values and the corresponding cuboidal MR patches centered on the voxels. Due to artifact corruption, some of these CT values may be incorrect; MAR can be viewed as the task of estimating the true CT values $\{y_i\}$, $\forall i \in \mathcal{T}$.

To achieve this, kerMAR models the probabilistic relationship between y_i, t_i and $\mathbf{m}_i$, in particular the *generative model*:

$$p(\{\mathbf{m}_i, y_i, t_i\}|\boldsymbol{\lambda}) = \prod_{i\in\mathcal{T}_i} p(\mathbf{m}_i, y_i, t_i|\boldsymbol{\lambda}) \quad \text{with}$$

$$p(\mathbf{m}_i, y_i, t_i|\boldsymbol{\lambda}) = p(t_i|y_i, \mathbf{m}_i, \boldsymbol{\lambda})p(y_i, \mathbf{m}_i|\boldsymbol{\lambda}).$$

where $\boldsymbol{\lambda}$ are the model hyperparameters. The joint distribution of CT values and MR patches is now learned from the uncorrupted patient data using kernel density estimation[2] with Gaussian kernels of precisions σ_y^2 and σ_m^2:

$$p(y_i, \mathbf{m}_i|\boldsymbol{\lambda}) = \frac{1}{|\mathcal{T}_u|}\sum_{n\in\mathcal{T}_u} \mathcal{N}(y_i|y_n, \sigma_y^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_n, \sigma_m^2\mathbf{I}_M),$$

where $\mathcal{T}_u$ is the set of uncorrupted voxels (defined as explained later).

The artifact noise model is chosen to be a Gaussian distribution independent of $\mathbf{m}_i$ with mean y_i and a voxel dependent variance $f_i\sigma_t^2$:

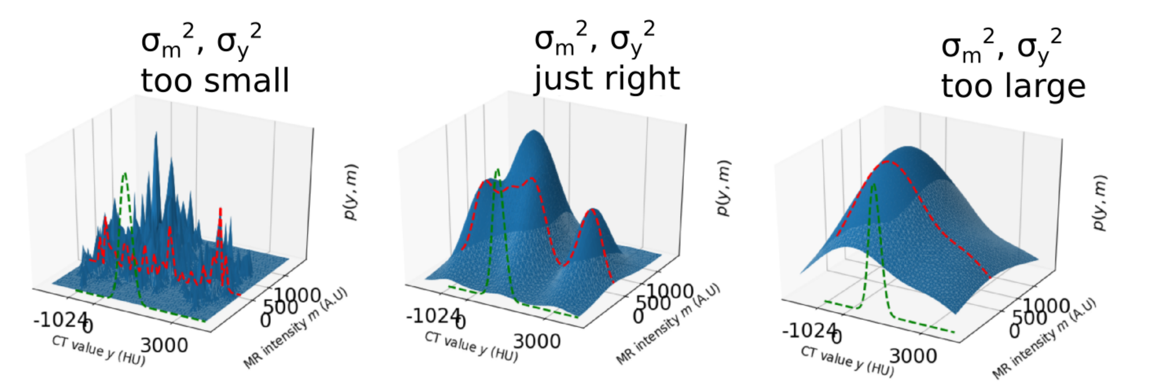
$$p(t_i|y_i, \boldsymbol{m}_i, \boldsymbol{\lambda}) = \mathcal{N}(t_i|y_i, f_i\sigma_t^2),$$

where f_i is a known function that scales the artifact noise variance over the image. We use a sigmoid function with range $[0;1]$ that decreases with distance to the metal implants, which is additionally used to define the set of uncorrupted voxels, $\mathcal{T}_u$, by truncating f_i such that $\mathcal{T}_u = [i \in \mathcal{T}|f_i \leq 0.5]$.

Thus knowing the generative model, all relevant distributions can be calculated. In particular, kerMAR combines using Bayes theorem the distributions $p(y_i, \mathbf{m}_i|\boldsymbol{\lambda})$ and $p(t_i|y_i, \boldsymbol{\lambda})$, respectively the surfaces and green curves on fig. 1 (here for simulated data and assuming 1x1x1 patches); this yields the posterior $p(y_i|\mathbf{m}_i, t_i)$ which, given an observed $\mathbf{m}_i$ and t_i as well as known hyperparameters $\boldsymbol{\lambda} = \{\sigma_t^2, \sigma_y^2, \sigma_m^2\}$, can be averaged over to estimate the true CT value. Graphically, this corresponds to multiplying the red and green curves on fig. 1 and averaging over the result; analytically it becomes the following weighted average:

$$\hat{y}_i = \int_{y_i} y_i p(y_i|\mathbf{m}_i, t_i)dy_i = \sum_{n\in\mathcal{T}} \mu_n^i v_n^i \quad \text{with} \quad \mu_n^i = \frac{\sigma_t^{-2}}{\sigma_t^{-2} + \sigma_y^{-2}}t_i + \frac{\sigma_y^{-2}}{\sigma_t^{-2} + \sigma_y^{-2}}y_n \tag{1}$$

$$\text{and} \quad v_n^i = \frac{\mathcal{N}(t_i|y_n, f_i\sigma_t^2 + \sigma_y^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_n, \sigma_m^2\mathbf{I}_M)}{\sum_{n'\in\mathcal{T}_u}\mathcal{N}(t_i|y_{n'}, f_i\sigma_t^2 + \sigma_y^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_{n'}, \sigma_m^2\mathbf{I}_M)} \tag{2}$$



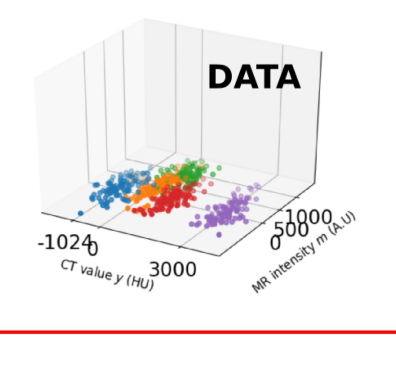


Figure 1:
The kerMAR model calculated on simulated data (left) using different parameter settings: Left: Too small kernel variances; middle: The kernel variances that generated the data; right: too large kernel variances.
The green curve illustrates the artifact noise model - the probability of a CT value is a compromise of this curve and the surface.

The kerMAR estimate thus becomes effectively a weighted average of a linear combination of the measured corrupted CT value t_i and the CT values in the uncorrupted patient volume, $\{y_n\}_{n\in\mathcal{T}_u}$.

Estimation of the hyperparameters (λ) using Empirical Bayes and Expectation Maximisation

For the kerMAR model to be self-contained, the hyperparameters $\boldsymbol{\lambda} = \{\sigma_y^2, \sigma_m^2, \sigma_t^2\}$ must be chosen in an informed manner. To understand how the choice of hyperparameters influences the kerMAR model, consider the illustration of $p(y_i, \mathbf{m}_i|\boldsymbol{\lambda})$ for three different parameter choices on fig. 1. The surfaces were calculated using the kerMAR model on simulated data; the middle surface in particular used the simulation parameters and is accordingly the more plausible description of the data. This motivates the use of our generative model to fit the parameters, which may be done by considering the marginal likelihood of the data given the hyperparameters. After some algebra, this becomes:

$$p(\{\mathbf{m}_i, t_i\}|\boldsymbol{\lambda}) = \prod_{i\in\mathcal{T}}\int_{y_i} p(\mathbf{m}_i, y_i, t_i|\boldsymbol{\lambda})dy_i \quad \text{with}$$

$$\int_{y_i} p(\mathbf{m}_i, y_i, t_i|\boldsymbol{\lambda})dy_i = \frac{1}{\mathcal{T}_u}\sum_{n\in\mathcal{T}_u}\mathcal{N}(t_i|y_n, \sigma_y^2 + f_i\sigma_t^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_n, \sigma_m^2\mathbf{I}_M),$$

Empirical Bayes (maximum likelihood) parameter estimation works by optimising this function, or, more tractably, its logarithm. This optimisation task is noticeably simplified and little affected in the end results by setting $f_i \to 0\ \forall i \in \mathcal{T}_u$ and 1 elsewhere, leading to the optimisation problem:

$$\arg\max_{\boldsymbol{\lambda}}\{\Phi(\boldsymbol{\lambda})\} \quad \text{with} \quad \Phi(\boldsymbol{\lambda}) = \sum_{i\in\mathcal{T}_u}\log\left(\sum_{n\in\mathcal{T}_u}\mathcal{N}(t_i|y_n, \sigma_y^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_n, \sigma_m^2)\right) + \sum_{i\notin\mathcal{T}_u}\log\left(\sum_{n\in\mathcal{T}_u}\mathcal{N}(t_i|y_n, \sigma_t^2 + \sigma_y^2)\mathcal{N}(\mathbf{m}_i|\mathbf{m}_n, \sigma_m^2)\right)$$

Off-the-shelf optimisation techniques such as gradient ascent or Newton's method could now be employed. However, to take advantage of the shape of the cost function and get significant speed gains, we opt to use Jensen's inequality ($\log(\sum_n v_n x_n) \geq \sum_n v_n \log(x_n)$ for $\sum_n v_n = 1$), to derive an Expectation Maximisation (EM) lower bound on the cost function that depends on a guess on the parameters $\boldsymbol{\lambda}^x$. This lower bound gains the following useful qualities: 1.) It consists of sums of logs rather than logs of sums, making it more tractable; and 2.) by the EM principle[1], maximising it is equivalent to maximising the true objective function. It looks as follows:

$$Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^x) = \sum_{i\in\mathcal{T}}\sum_{n\in\mathcal{T}_u} v_n^i \log(\mathcal{N}(i|\mathbf{m}_n, \sigma_m^2\mathbf{I}_M)) + \sum_{i\in\mathcal{T}_u}\sum_{n\in\mathcal{T}_u} v_n^i \log(\mathcal{N}(t_i|y_n, \sigma_y^2)) + \sum_{i\notin\mathcal{T}_u}\sum_{n\in\mathcal{T}_u} v_n^i \log(\mathcal{N}(t_i|y_n, \sigma_t^2 + \sigma_y^2)),$$

where the EM-weights $v_n^{i\,(x)}$ are calculated at the hyper-parameter estimate at iteration x using eqn. (2). The EM-algorithm starts from an initial guess $\boldsymbol{\lambda}^0$ that is iteratively updated by alternately calculating the weights (the E-step) and maximising the lower bound (the M-step), which can be done in closed form.

Results and Discussion

We show on fig. 2, along with the uncorrected Filtered Back Projections (FBP) on which they were calculated, the kerMAR images using the estimated parameters ($\boldsymbol{\lambda}$) as well as scaled to be respectively lower and higher. This emulates the situation on fig. 1 (top-down on fig. 3 corresponding to right-left on fig. 1). The red arrows point to regions where the empirically chosen parameters apparently outperformed the scaled versions, while the blue arrows show cases where gains could be had by using larger variances. While the assignment of the arrows is somewhat subjective, the large ratio of red to blue arrows along with the consistently good performance when using the estimated $\boldsymbol{\lambda}$ is encouraging for the model, as this implies that the model fits the data well and may be used in a self-contained fashion without further adjustment of hyperparameters.

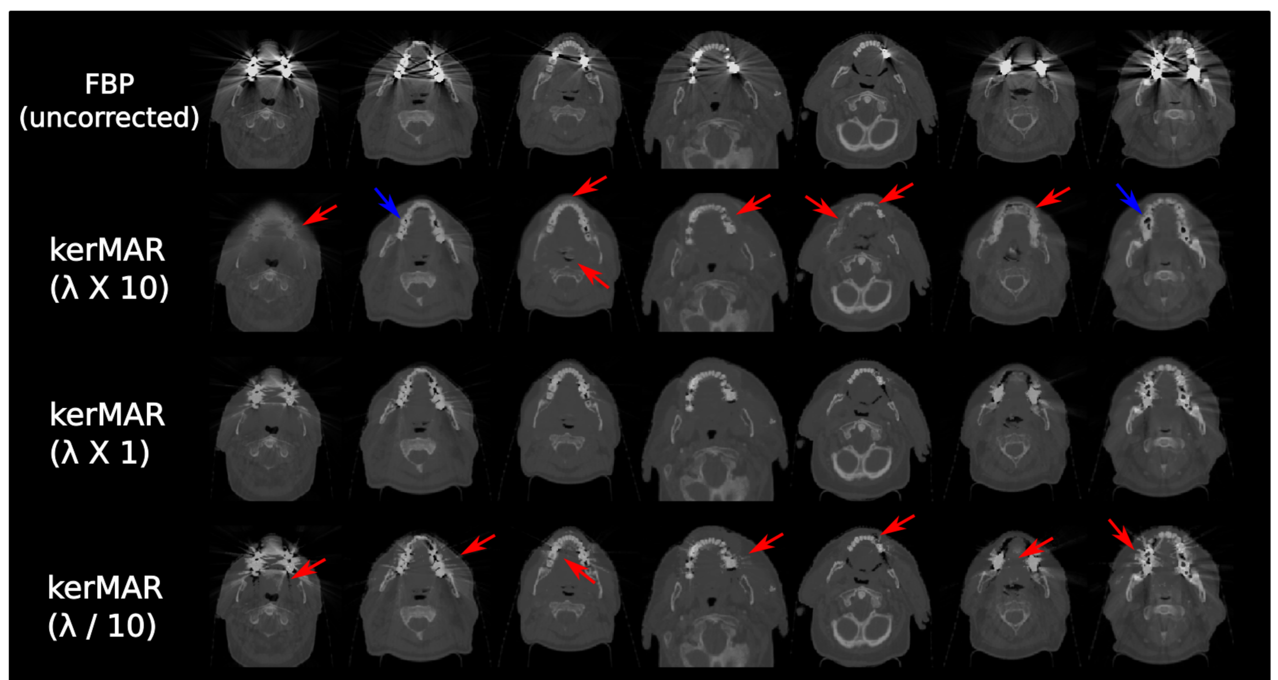


Figure 2:
kerMAR calculated for the estimated parameters (third row) as well as scaled with respectively a factor of 10 and 1/10 (second and fourth rows). Top row shows the original FBPs. Red arrows point to regions where the empirically chosen parameters worked better, blue arrows to where the factor led to apparent improvements.

In addition to providing apparent improves over the scaled parameter versions, the use of automatically tuned hyperparameters is of great utility both when transferring the application of the model to different image sets and when using the model as a part of more complex algorithms with added parameters, into which it may fit without having to retune anything. An example is Maximum A Posteriori CT reconstruction, in which the distribution $p(y_i|\mathbf{m}_i, t_i)$ may readily be used as a prior.

Conclusions

We have presented the kerMAR generative model and inference algorithm for metal artifact reduction, and considered tuning its hyperparameters Empirical Bayes and the EM algorithm. Using the thusly chosen parameters with kerMAR led to improved artifact reduced images than using scaled versions of the parameters, underlying both the feasiblity and utility of tuning the hyperparameters on the data.

References

[1] Thomas P. Minka. Expectation-maximization as lower bound maximization. https://tminka.github.io/papers/minka-em-tut.pdf, 2009.

[2] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.