Collisionality dependence of impurity transport in Alcator C-Mod H-modes

M.A. Chilenski, M. Greenwald, N.T. Howard, M.L. Reinke,[1]
J.E. Rice, A.E. White, Y. Marzouk[2]

MIT PSFC/Alcator C-Mod
[1]University of York
[2]MIT Aero/Astro, Uncertainty Quantification Group

October 29, 2014

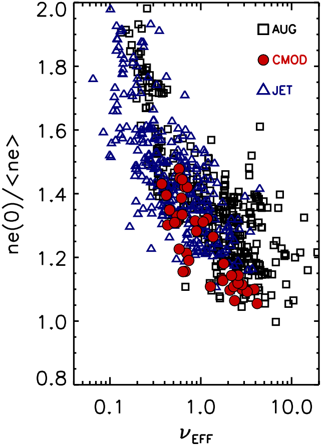
Supported by USDoE award DE-FC02-99ER54512. Supported in part by a DOE SCGF fellowship, administered by ORISE-ORAU under contract DE-AC05-06OR23100.

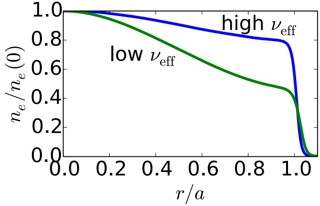
Recent experiments have been performed to collect a data set over wide range of collisionality, explore collisionality dependence of H-mode impurity transport

Outline

- Electron density peaking in H-mode
- Initial look at global confinement trends
- Measuring impurity transport coefficients on C-Mod
- New analysis/uncertainty quantification techniques
- Initial look at core transport coefficient profiles

H-mode electron density peaking scales with collisionality





Cross-machine scaling law [1, 2]

$n_e(0)/\langle n_e \rangle$ scales with $\nu_{eff} = 0.1 Z_{eff} R \frac{\langle n_{e,19} \rangle}{\langle T_{e,keV} \rangle^2}$

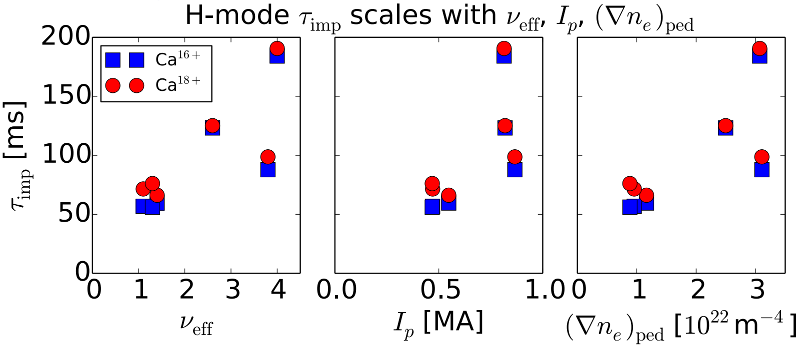
This scaling enables interesting studies of impurity transport

- Explore connection between electron and impurity peaking.
- Actuate the neoclassical pinch and turbulence drive (both $\propto \nabla n_e$), explore implications for high-Z accumulation [3].

[1] Angioni et al. (2007), POP [2] Greenwald et al. (2007), NF

[3] Reinke et al., JI1.00002 (tuesday PM)

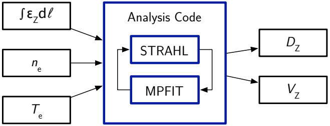
Initial look at global confinement

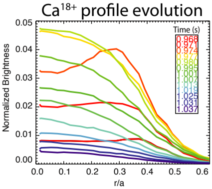


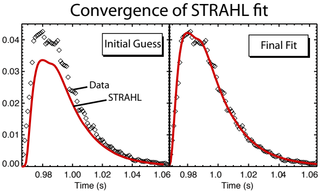
- All discharges are EDA H-modes: no ELMs, pedestal regulated by quasi-coherent mode (QCM).
- Scanned I_p to vary $\langle n_e \rangle$, scanned P_{ICRF} to vary $\langle T_e \rangle$.
- Strong scaling with $(\nabla n_e)_{\mathrm{ped}}$ consistent with J. Rice, PO3.00004 (earlier this session).
- Challenge: extract local core transport information from plasma with global confinement dominated by pedestal.

Alcator C-Mod is uniquely equipped to make detailed measurements of impurity transport

- Multipulse laser blow-off impurity injector can inject Ca (or other desired impurity) multiple times per shot [1].
- X-ray imaging crystal spectrometer [2] observes spatial profile of a *single charge state* (Ca^{18+}): more direct interpretation than unresolved soft x-rays.
- Impurity transport coefficient profiles reconstructed [3] using STRAHL impurity transport code [4].
- *Result is strongly sensitive to background n_e, T_e profiles.*







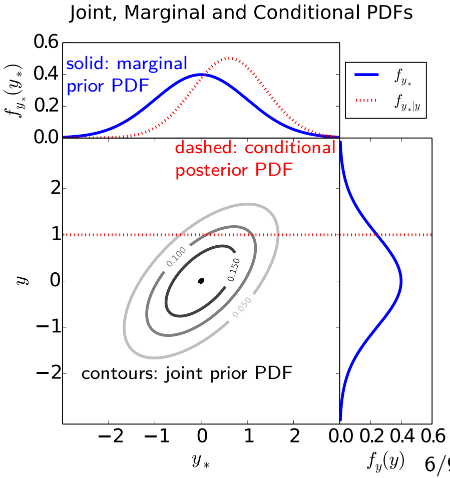
[1] Howard et al. (2011), RSI [2] Ince-Cushman et al. (2008), RSI
[3] Howard et al. (2012), NF [4] Dux (2006), IPP rpt. 10/30

New techniques for uncertainty propagation

- Profile fitting is fundamental to transport analysis.
- Splines are bad at finding uncertainty in gradient.
- New approach [1]: Gaussian process regression (GPR) [2]
  - Automatic, delivers reliable uncertainty estimate for gradient.
  - Based on spatial covariance of data points, *not* an assumed functional form.
  - Output is *probability distribution* (PDF), can be efficiently propagated through subsequent analysis.

y: observation, y_*: prediction

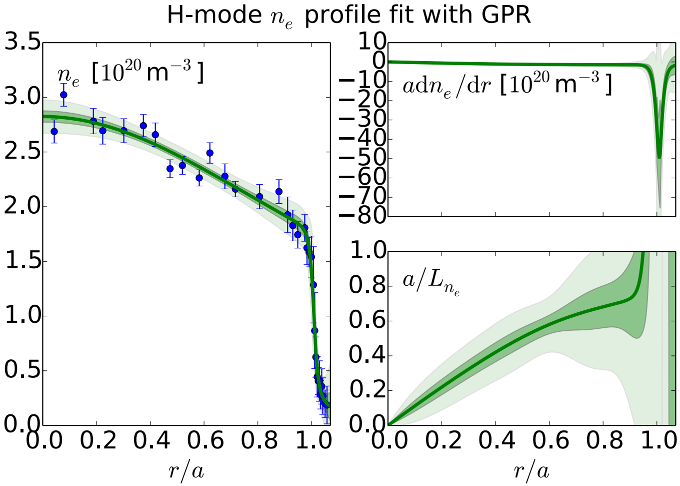
$$\underbrace{f(y_*|y=1)}_{\substack{\text{conditional posterior} \\ \text{“PDF of } y_* \text{ given } y\text{”}}} = \frac{\overbrace{f(y_*, y=1)}^{\text{joint prior}}}{\underbrace{f(y=1)}_{\substack{\text{marginal prior} \\ f(y)=\int f(y_*,y)\,\mathrm{d}y_*}}}$$



[1] Chilenski et al. (2014), submitted to NF. Preprint: PSFC report JA-14-22

[2] Rasmussen and Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006

Example: Fitting an H-mode density profile

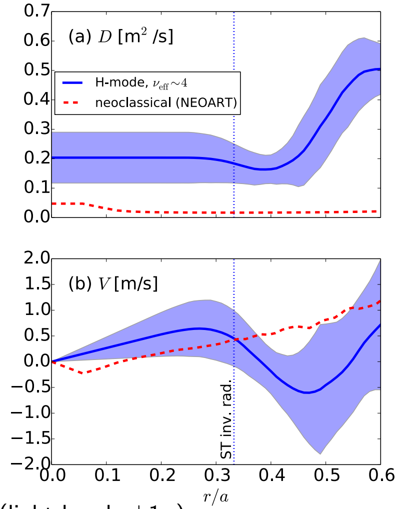


(dark band: $\pm 1\sigma$, light band: $\pm 3\sigma$)

Open source software available: github.com/markchil/gptools/

Preliminary analysis yields H-mode core impurity transport coefficient profiles

PRELIMINARY STRAHL Results



(light band: $\pm 1\sigma$)

- Reconstruction not reliable for $r/a > 0.6$.
- High D inside inversion radius could simply be result of incomplete handling of sawteeth, further analysis is necessary.
- Core impurity transport dominated by anomalous diffusivity.

Alcator C-Mod's combination of powerful diagnostics, novel analysis techniques and reactor-relevant conditions enables detailed study of H-mode impurity transport

Summary

- Scaling of electron density peaking with ν_{eff} provides convenient actuator for a/L_{n_e}: vary neoclassical pinch and turbulence drive.
- Have made detailed measurements of impurity transport in EDA H-modes of varying ν_{eff}.
- Reconstruction of D, V profiles has begun.
- New data analysis techniques have been developed to fit profiles, efficiently propagate uncertainty.

Future Work

- Analyze more shots with different collisionalities.
- Compare to gyrokinetic simulations.

Backup slides

1. Gaussian processes
2. gptools
3. Bayesian inference

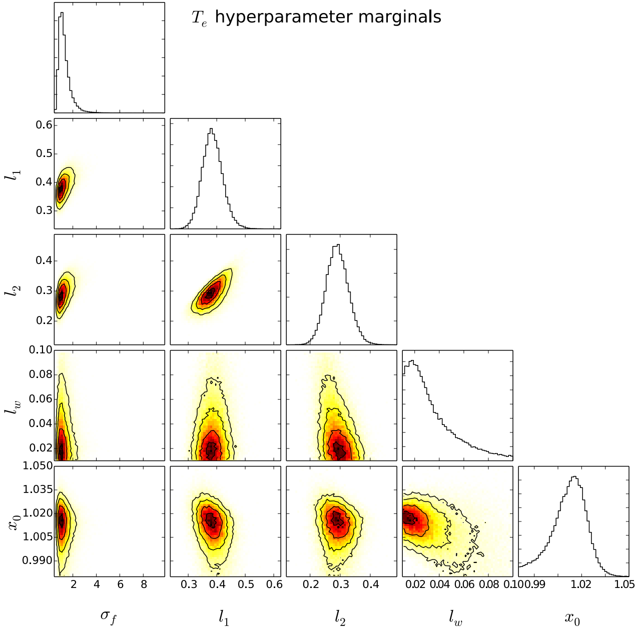
The hyperparameters can be estimated by maximizing the likelihood

Likelihood of the training data given k with hyperparameters $\boldsymbol{\theta} = [\sigma_f,\ \ell,\ \ldots]$:

$$\ln p(\boldsymbol{y}|\mathrm{X},\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}[\mathrm{K}+\Sigma_{\mathrm{n}}]^{-1}\boldsymbol{y} - \frac{1}{2}\ln|\mathrm{K}+\Sigma_{\mathrm{n}}| - \frac{n}{2}\ln 2\pi$$

- Maximize with respect to hyperparameter vector $\boldsymbol{\theta}$.
- Local maxima: different possible interpretations of the data. E.g., noisy and long-ℓ versus precise and short-ℓ
- Compare likelihoods to select the most appropriate kernel.

Full treatment of hyperparameters uses MCMC integration

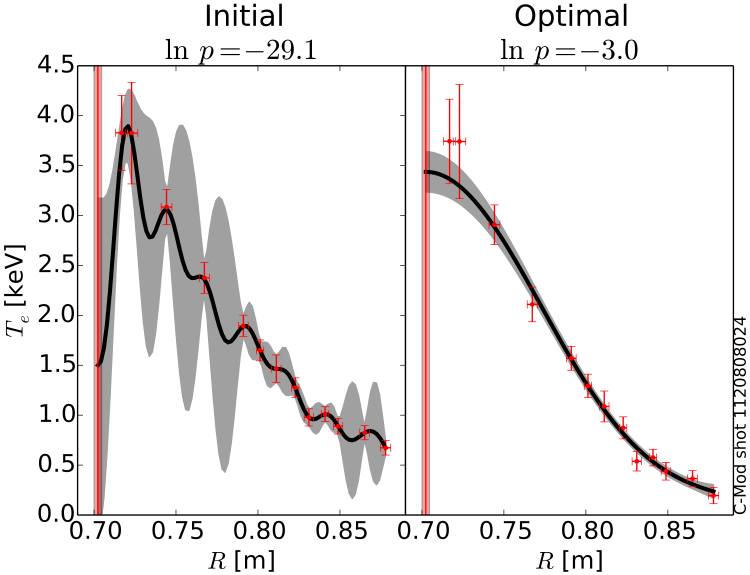


Point estimate misses substantial uncertainty in gradient

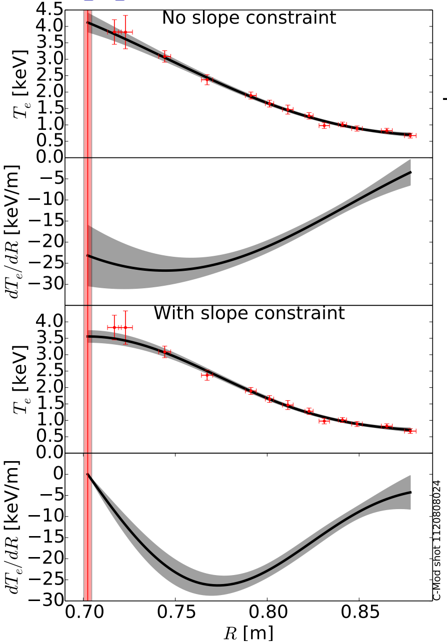
Median relative uncertainties over $0 \leq \psi_n \leq 1$

Quantity	y	y'	a/L_y
n_e, MAP	1.2%	6.0%	6.0%
n_e, MCMC	1.4%	8.4%	8.3%
T_e, MAP	1.3%	3.7%	4.0%
T_e, MCMC	1.4%	5.4%	5.6%

Bad versus good choices for the hyperparameters have a large effect on the likelihood



Getting gradients *and their uncertainties* is straightforward



The derivative of a GP is a GP:

$$\text{cov}\left(y_i,\, \frac{\partial y_j}{\partial x_{dj}}\right) = \frac{\partial k(\boldsymbol{x}_i,\, \boldsymbol{x}_j)}{\partial x_{dj}}$$

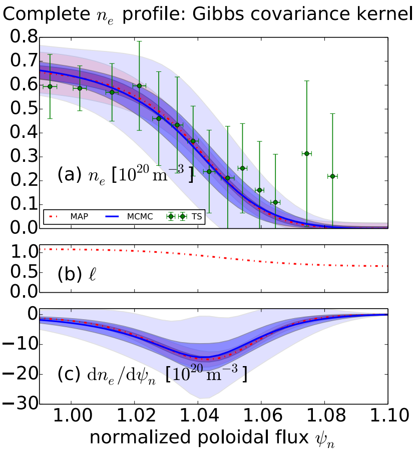
$$\text{cov}\left(\frac{\partial y_i}{\partial x_{di}},\, \frac{\partial y_j}{\partial x_{dj}}\right) = \frac{\partial^2 k(\boldsymbol{x}_i,\, \boldsymbol{x}_j)}{\partial x_{di}\,\partial x_{dj}}$$

- Derivative equality constraint: just add a datapoint!
- Derivative predictions: predictive distribution contains the uncertainty.

Capturing the pedestal requires a non-stationary kernel

Gibbs kernel: ℓ is an arbitrary function of $\boldsymbol{x}$

$$k_G(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(\frac{2\ell(\boldsymbol{x})\ell(\boldsymbol{x}')}{\ell^2(\boldsymbol{x}) + \ell^2(\boldsymbol{x}')} \right)^{1/2} \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x}'|^2}{\ell^2(\boldsymbol{x}) + \ell^2(\boldsymbol{x}')} \right)$$

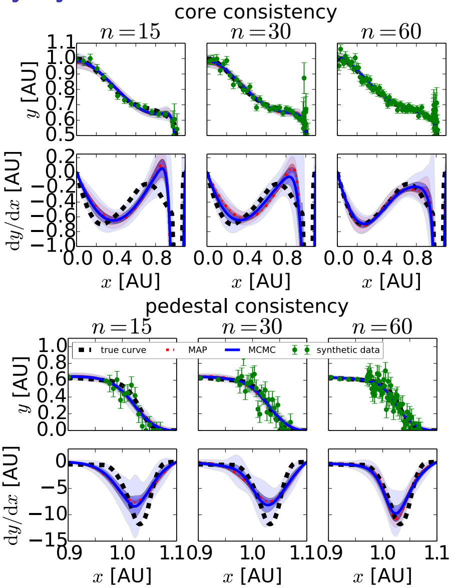


- Length scale:

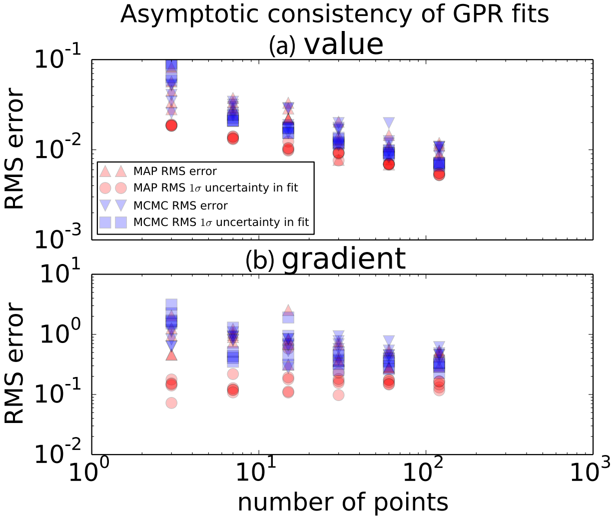
$$\ell = \frac{\ell_1 + \ell_2}{2} - \frac{\ell_1 - \ell_2}{2} \tanh \frac{x - x_0}{\ell_w}$$

- Handled ℓ_1, ℓ_2, ℓ_w and x_0 by maximizing $\ln p$ (MAP) and by marginalizing with MCMC.

Gibbs kernel with tanh length scale has been extensively tested with noisy synthetic data



Error estimates from Gibbs kernel with tanh length scale have been shown to be asymptotically consistent using noisy synthetic data



gptools: An extensible, object-oriented Python package for multivariate GPR including gradients

- Available GPR codes lack one or more critical features:
  - Ability to both constrain and predict gradients.
  - Straightforward way to draw random samples.
- gptools was written to meet these needs:
  - Object-oriented structure.
  - Interface for easy data fusion and application of constraints.
  - SE, Gibbs, Matérn and RQ kernels *with support for arbitrary orders of differentiation*.
- Available on GitHub: www.github.com/markchil/gptools

gptools contains two classes for performing GPR

GaussianProcess

k : Kernel
nk : Kernel
X
n
y
err_y

add_data(X, y, err_y, n)
optimize_hyperparameters()
predict(X_star)
draw_sample(X_star)

Kernel

num_dim
params
fixed_params
param_bounds

__call__(Xi, Xj, ni, nj)
set_hyperparams(new_params)

gptools: An extensible, object-oriented Python package for multivariate GPR including gradients

```python
import gptools

# Create kernel:
k = gptools.SquaredExponentialKernel(1)
# Create GP:
gp = gptools.GaussianProcess(k, X=R_mid, y=Te, err_y=err_Te)
# Impose zero slope constraint at magnetic axis:
gp.add_data(R_mag, 0, n=1)
# Optimize hyperparameters:
gp.optimize_hyperparameters()

# Make a prediction of the value:
R_star = scipy.linspace(R_mag, R_mid.max(), 100)
Te_fit, Te_std = gp.predict(R_star)
# Make a prediction of the gradient:
gradTe_fit, gradTe_std = gp.predict(R_star, n=1)
```

gptools implements a very general form of GPR

$$f\left(\begin{bmatrix}\boldsymbol{M}_*\\ \boldsymbol{M}\end{bmatrix}\right)=\mathcal{N}\Bigg(\begin{bmatrix}\mathrm{T}_* & 0\\ 0 & \mathrm{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{\mu}(\mathrm{X}_*)\\ \boldsymbol{\mu}(\mathrm{X})\end{bmatrix},$$
$$\begin{bmatrix}\mathrm{T}_* & 0\\ 0 & \mathrm{T}\end{bmatrix}\begin{bmatrix}\mathrm{K}(\mathrm{X}_*,\mathrm{X}_*) & \mathrm{K}(\mathrm{X},\mathrm{X}_*)\\ \mathrm{K}(\mathrm{X}_*,\mathrm{X}) & \mathrm{K}(\mathrm{X},\mathrm{X})\end{bmatrix}\begin{bmatrix}\mathrm{T}_*^T & 0\\ 0 & \mathrm{T}^T\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & \Sigma_M\end{bmatrix}\Bigg)$$

$$\ln\mathcal{L}=-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln\left|\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M\right|$$
$$-\frac{1}{2}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X}))^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X}))$$

$$f(\boldsymbol{M}_*|\boldsymbol{M})=\mathcal{N}\big(\mathrm{T}_*\boldsymbol{\mu}(\mathrm{X}_*)+\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X})\mathrm{T}^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}(\boldsymbol{M}-\mathrm{T}\boldsymbol{\mu}(\mathrm{X})),$$
$$\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X}_*)\mathrm{T}_*^T-\mathrm{T}_*\mathrm{K}(\mathrm{X}_*,\mathrm{X})\mathrm{T}^T(\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X})\mathrm{T}^T+\Sigma_M)^{-1}\mathrm{T}\mathrm{K}(\mathrm{X},\mathrm{X}_*)\mathrm{T}_*^T\big)$$

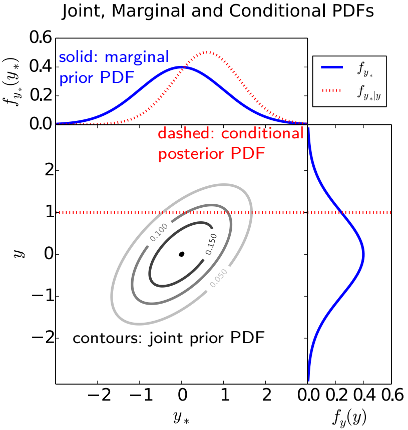
- Supports data of arbitrary dimension $\boldsymbol{x}\in\mathbb{R}^n$.
- Supports explicit, parametric mean function $\mu(\boldsymbol{x})$: can perform nonlinear regression with GP fit to residuals.
- Supports arbitrary linear transformations T, T_* of inputs, outputs: can perform tomographic inversions constrained with point measurements.
- Supports noise of arbitrary structure Σ_M on observations.

GPR: a probabilistic method to fit profiles

$$\underbrace{f(y_*|y=1)}_{\substack{\text{conditional posterior} \\ \text{“PDF of } y_* \text{ given } y\text{”}}} = \frac{\overbrace{f(y_*, y=1)}^{\text{joint prior}}}{\underbrace{f(y=1)}_{\substack{\text{marginal prior} \\ f(y)=\int f(y_*,y)\,dy_*}}}$$

- Create multivariate normal *prior distribution* that sets smoothness, symmetry, etc.
- *Condition* on observations to yield the fit, including uncertainty estimate.
- Distribution can include derivatives, line integrals, volume averages, etc.

y: observation, y_*: prediction



(PDF = “probability density function”)

The properties of the fit are inferred using Bayes' rule

Given spatial covariance characterized by hyperparameter vector $\boldsymbol{\theta}$ and observations $\boldsymbol{y}$, Bayes' rule is:

$$\underbrace{f(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{f(\boldsymbol{y}|\boldsymbol{\theta})}^{\text{likelihood}}\ \overbrace{f(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{f(\boldsymbol{y})}_{\text{evidence}}}$$

- **Likelihood**: Probability of observing the data $\boldsymbol{y}$ given the hyperparameters $\boldsymbol{\theta}$. Simply a multivariate normal for GPR.
- **Prior**: Distribution encoding any prior assumptions about the hyperparameters $\boldsymbol{\theta}$ (positivity, typical values, etc.)
- **Evidence**: Probability of the data under the model. Just a normalization constant for parameter estimation.
- **Posterior**: Probability distribution for the hyperparameters $\boldsymbol{\theta}$ given the data $\boldsymbol{y}$: the end-goal of the inference.

Three levels of sophistication to select hyperparameters $\boldsymbol{\theta}$

$$\underbrace{f(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{f(\boldsymbol{y}|\boldsymbol{\theta})}^{\text{likelihood}}\ \overbrace{f(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{f(\boldsymbol{y})}_{\text{evidence}}}$$

- so-so: **Maximum likelihood** (ML): Pick the hyperparameters $\boldsymbol{\theta}$ that maximize the **likelihood** $f(\boldsymbol{y}|\boldsymbol{\theta})$ of the data.
- better: **Maximum a posteriori** (MAP, "empirical Bayes"): Pick the hyperparameters $\boldsymbol{\theta}$ that have the highest **posterior probability** $f(\boldsymbol{\theta}|\boldsymbol{y})$.
- best: **Marginalization** ("full Bayes"): Average over the possible hyperparameters when making a prediction $\boldsymbol{y}_*$:

$$f(\boldsymbol{y}_*|\boldsymbol{y}) = \int f(\boldsymbol{y}_*|\boldsymbol{y}, \boldsymbol{\theta}) f(\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta}$$