Skip to content

Building Intelligent Probabilistic Systems

Machine learning, statistics, neuroscience, everything…

- About
- Log in
- Register

Search

Categories:

- Compression
- Computation
- Machine Learning
- Meta
- Neuroscience
- Probability
- Ramblings
- Recent work
- Statistics
- Uncategorized

Fisher information

I first heard about Fisher information in a statistics class, where it was given in terms of the following formulas, which I still find a bit mysterious and hard to reason about:

$$\begin{aligned}\mathbf{F}_\theta &= \mathbb{E}_x[\nabla_\theta \log p(x;\theta) (\nabla_\theta \log p(x;\theta))^T] \\ &= \mathrm{Cov}_x[\nabla_\theta \log p(x;\theta)] \\ &= \mathbb{E}_x[-\nabla^2_\theta \log p(x;\theta)].\end{aligned}$$

It was motivated in terms of computing confidence intervals for your maximum likelihood estimates. But this sounds a bit limited, especially in machine learning, where we're trying to make predictions, not present someone with a set of parameters. It doesn't really explain why Fisher information seems so ubiquitous in our field: natural gradient, Fisher kernels, Jeffreys priors, and so on.

This is how Fisher information is generally presented in machine learning textbooks. But I would choose a different starting point: Fisher information is the second derivative of KL divergence.

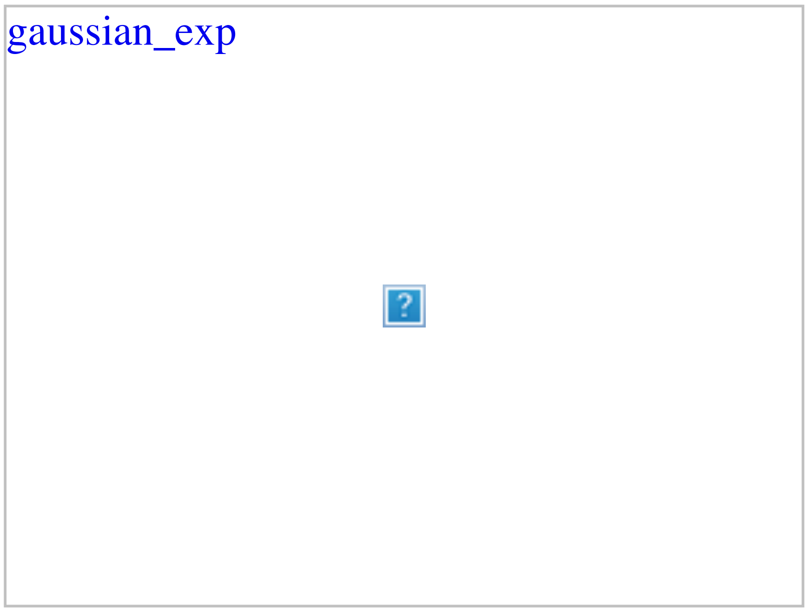
$$\begin{aligned}\mathbf{F}_\theta &= \nabla^2_{\theta'} \mathrm{D}(\theta' \| \theta)|_{\theta'=\theta} \\ &= \nabla^2_{\theta'} \mathrm{D}(\theta \| \theta')|_{\theta'=\theta}\end{aligned}$$

(Yes, you read that correctly — both directions of KL divergence have the same second derivative at the point where the distributions match, so locally, KL divergence is approximately symmetric.) In other words, by taking the second-order Taylor expansion, we can approximate the KL divergence between two nearby distributions with parameters θ and θ' in terms of Fisher information:

$$\mathrm{D}(\theta' \| \theta) \approx \frac{1}{2}(\theta' - \theta)^T \mathbf{F}_\theta (\theta' - \theta).$$

Since KL divergence is roughly analogous to a distance measure between distributions, this means Fisher information serves as a local distance metric between distributions. It tells you, in effect, how much you change the distribution if you move the parameters a little bit in a given direction.

This gives us a way of visualizing Fisher information. In the following figures, each of the ovals represents the set of distributions which are distance 0.1 from the center under the Fisher metric, i.e. those distributions which have KL divergence of approximately 0.01 from the center distribution. I'll refer to these as "Fisher balls." Here is the visualization for a univariate Gaussian, parameterized in terms of mean μ and standard deviation σ:



This visualization shows that when σ is large, changing the parameters has less effect on the distribution than when σ is small. We can also repeat this visualization for the information form representation,

$$p(x; h, \lambda) \propto \exp\left(-\frac{\lambda}{2}x^2 + hx\right).$$

Why do the ovals fan out? Intuitively, if you rescale h and λ by the same amount, you're holding the mean fixed, so the distribution changes less than it would if you varied them independently.

Wouldn't it be great if we could find some parameterization where all the Fisher balls are unit circles? Unfortunately, there's generally no way to enforce this globally. However, we can enforce it locally by applying an affine transformation to the parameters:

$$\eta - \eta_0 = \mathbf{F}_{\theta_0}^{1/2}(\theta - \theta_0). \tag{1}$$

This stretches out the parameter space in the directions of large Fisher information and shrinks it in the directions of small Fisher information. Then KL divergence looks like squared Euclidean distance near η_0:

$$\mathrm{D}(\eta \| \eta_0) \approx \mathrm{D}(\eta_0 \| \eta) \approx \frac{1}{2}\|\eta - \eta_0\|^2.$$

What's nice about this representation is that the local properties no longer depend on the parameterization. I.e., no matter what parameterization θ you started with, the transformed space looks roughly the same near η_0, up to a rigid transformation. This gives a way of constructing mathematical objects that are invariant to the parameterization. Roughly speaking, if an algorithm is defined in terms of local properties of the model (such as gradients), you can apply the same algorithm in the transformed space, and it won't depend on the parameterization.

When we think about Fisher information in this way, it gives some useful intuitions for why it appears in so many places:

1. As I mentioned above, Fisher information is most commonly motivated in terms of the asymptotic variance of a maximum likelihood estimator. I.e.,

   $$\hat{\theta} \sim \mathcal{N}(\theta, \frac{1}{N}\mathbf{F}_\theta^{-1}),$$

   where N is the number of data points. But what this is really saying is that if you transform the space according to (1), the maximum likelihood estimate is distributed as a spherical Gaussian with standard deviation $1/\sqrt{N}$.

2. Natural gradient [1] is a variant of stochastic gradient descent which accounts for curvature information. It basically works by stretching the space according to (1), and computing the gradient in the transformed space. This is analogous to Newton's method, which essentially computes the gradient in a space that's stretched according to the Hessian. Riemannian manifold HMC [2] is an MCMC sampler based on the same idea.
3. The Jeffreys prior is an "uninformative" prior over the parameters of a probability distribution, defined as:

   $$p(\theta) \propto \sqrt{\det \mathbf{F}_\theta}.$$

   Since the volume of a Fisher ball is proportional to $1/\sqrt{\det \mathbf{F}_\theta}$, this distribution corresponds to allocating equal mass to each of the Fisher balls. The virtue is that the prior doesn't depend on how you parameterized the distribution.

4. Minimum message length [3] is a framework for model selection, based on compression. From a model you construct a two-part code for a dataset: first you encode the model parameters (using some coding scheme), and then you encode the data given the parameters. If you don't want to assume anything about the process generating the data, you might choose a coding scheme which minimizes the regret: the number of extra bits you had to spend, relative to if you were given the optimal model parameters

in advance. If the parameter space is compact, an approximate regret-minimizing scheme basically involves tiling the space with K Fisher balls (for some K), using log K bits to identify one of the balls, and coding the data using the parameters at the center of that ball. In the worst case, the data distribution lies at the boundary of the ball. Interestingly, this scheme has a very similar form to the Jeffreys prior, but comes from a very different motivation.

5. Information geometry [4] is a branch of mathematics that uses differential geometry to study probabilistic models. The goal is to analyze spaces of probability distributions in terms of their intrinsic geometry, rather than by referring to some arbitrary parameterization. Defining a Riemannian manifold requires choosing a metric, and for a manifold of probability distributions, that metric is generally Fisher information.

[1] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 1998.

[2] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 2011.

[3] C. S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer, 2005.

[4] S. Amari and H. Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2007.

Posted in Statistics.

1 comment

By Roger Grosse – April 8, 2013

One Response

Stay in touch with the conversation, subscribe to the RSS feed for comments on this post.

1. *sorcererofdm says*

   This post helped a lot clarifying the reason why fisher information is chosen as the standard metric in IG. It’d be better if the pictures aren’t missing.

   September 1, 2015, 9:55 am Log in to Reply

You must be logged in to post a comment.

« The Gumbel-Max Trick for Discrete Distributions Optimal Spatial Prediction with Kriging »

Subscribe

Site RSS Feed

About Building Intelligent Probabilistic Systems

Welcome to the blog of the Harvard Intelligent Probabilistic Systems group! Here, members of HIPS as well as guest bloggers post about interesting research in machine learning, statistics, artificial intelligence, theoretical neuroscience, and related areas. Our goal is to share important work and ideas, from both past and present, that will aid in the development [...]more →

Search for: Search

Recent Posts

- Harvard Center for Research on Computation and Society: Call for Fellows and Visiting Scholars
- Which research results will generalize?
- Prior knowledge and overfitting
- ICML Highlight: Fast Dropout Training
- Testing MCMC code, part 2: integration tests

Recent Comments

- mattismyname on The Gumbel-Max Trick for Discrete Distributions
- sorcererofdm on Fisher information
- sitaram on Computing Log-Sum-Exp
- canonicalform on The Fundamental Matrix of a Finite Markov Chain
- jonas.wallin on Testing MCMC code, part 1: unit tests

Archives

- October 2014
- September 2014
- August 2013
- June 2013
- May 2013
- April 2013
- March 2013
- February 2013
- January 2013
- December 2012

Categories

- Compression
- Computation
- Machine Learning
- Meta
- Neuroscience
- Probability
- Ramblings
- Recent work
- Statistics
- Uncategorized

Blogroll

- Andrew Gelman
- Computational Complexity
- Daniel Lemire
- Hal Daumé III
- Harry Lewis
- Inducto Ex Machina
- Larry Wasserman
- Learning in Vision
- Machine Learning (Theory)
- Mathematics and Computation
- Michael Mitzenmacher
- My Slice of Pizza
- Nuit Blanche
- Radford Neal
- Scott Aaronson
- Talking Brains
- Terry Tao
- The Geomblog
- This Number Crunching Life
- Timothy Gowers
- xcorr

Meta

- Register
- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Tags

approximate inference back-propagation basic concepts Bayesian networks Bayesian nonparametrics belief-propagation cognitive biases computational complexity connectomics crowdsourcing deep density model deep learning empirical methods energy landscape exponential families feature learning graphical models Hamiltonian Monte Carlo hashing history inference information theory learning learning theory linear algebra markov Markov chains maximum entropy MCMC model selection Monte Carlo NIPS Normal distributions online learning representation sketching sparsity statistical modeling stochastic variational inference streaming supervised learning tips-and-tricks unsupervised learning vision visualization

---