Measuring $\mathcal{R}(D^{(*)})$ for $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$ using Semileptonic Tags and τ Decays to Hadrons

Thesis by
Daniel Chao

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2018
Defended 9/8/2016

Acknowledgements

To my advisors Prof. Frank Porter and Prof. David Hitlin - Thank you for supporting all of the ideas that I've tried and for having the patience to see them through. Thank you for being a role model in scientific integrity, as well as for setting the high standards for research quality. Finally, thank you for putting up with my naiveté and impatience.

To our group member Jake - Thank you for carrying this analysis over the last year and a half, and especially for your work in driving the analysis of the systematic uncertainties. Without your effort and insights, this analysis would have never been completed. Thank you for pushing this analysis through the RC and for taking their heat on my behalf; I am grateful for your patience towards them even when it appeared to drag on at times.

Again to Frank and Jake - Thank you for allocating every Thursday night over the last year and a half to hop on the phone with me. Despite the ups and downs and the heated arguments during this process, I am truly thankful for your patience towards me and for the sacrifices you made on my behalf.

Again to Dave - Thank you for your patience towards this analysis since the beginning, but especially over the last year and a half during my absence. I am eternally grateful of your pardon over this arrangement, and for the additional resources you had to allocate and sacrifice on my behalf.

To our other group members, especially Bertrand Echenard, Chih-hsiang Cheng, Piti Ongmongkolkul, and Yunxuan Li - Thank you for all your help and for your encouragements. This project would not have gone as smoothly or timely if it were not for your efforts.

To family, and friends - without whom, nothing is possible.

Abstract

We perform a measurement of $\mathcal{R}(D^{(*)})$ for $B \to \overline{D}^{(*)}\tau\nu_\tau$ using semileptonic tagging and τ decays to hadrons on the 429 fb^{-1} of data that *BABAR* collected at the $\Upsilon(4S)$ resonance. This is the first measurement of $\mathcal{R}(D^{(*)})$ using the specified reconstruction channels. Candidate selection was performed with supervised learning, where the training labels were obtained by solving an instance of subgraph isomorphism. The signal extraction was performed by solving an optimization problem whose objective function required the evaluation of kernel density estimates that were accelerated by a branch-and-bound algorithm as well as with a GPU. The training data for the density estimates were themselves the output of two classifier scores. We present a 68% and 95% confidence regions of $\mathcal{R}(D^{(*)})$, which do not show enough evidence to reject the standard model prediction.

Contents

Chapter 1

Introduction

This chapter tells the high level backstory of this project. It is meant to motivate the questions that we are primarily interested in without going into the details that experts might be attentive to. Any relevant details that are skipped in this fashion will be addressed in the appropriate chapter later in this document, and we direct the expert reader to the appropriate place whenever possible.

1.1 Background

This section introduces some of the elementary terminology and concepts that might be useful for those without much background in the field. It aims to give just enough information for the purpose of understanding the motivation, the methods, and the results of this project.

1.1.1 The Standard Model

The Standard Model (SM) is a mathematical model that physicists have constructed to describe the fundamental laws of nature. Actually, it is more appropriate to say that it is our best description of subatomic particles, consistent with all of our experimental observations known to date. Indeed, the goal of a typical high energy physicist is often to find ways to reveal deficiencies within the Standard Model, leading to, hopefully, hints of the unknown.

It might be unconvincing, and at best surprising, that all complicated entities and phenomena in nature can be reduced to just the Standard Model. Indeed, just as in civil planning, understanding how a brick is manufactured does not necessarily imply an understanding of how they can be assembled into an entire city. Nonetheless, a typical high energy physicist tends to be interested in this type of reductionism, and it just so happens that many of them agree that the Standard Model is, at present, the inner most layer of their understanding.

While there is only one equation that describes the Standard Model, there are various ways in which one can focus their descriptions and introductions. In the present context, it is useful to consider the Standard Model as a model that describes nature entirely in terms of matter and their interactions. More than that, all matter, and even interactions themselves, are modelled as the creation and annihilation of certain elementary particles. These particles are listed below for completeness:

- Quarks: u, d, s, c, b, and t.
- Leptons: e (electron), μ (muon), τ, ν_e, ν_μ, and ν_τ[1].
- Gauge Bosons: γ (photon), g (gluon), W, and Z.
- H (Higgs boson).

[1]The ν particles are also referred to as neutrinos.

1.1.2 Particle physics

The myriad of particle names can sometimes be perplexing, since the nomenclature might not seem to follow any rhyme or reason. This is due to the historical development of the field, and one can view it as simply that. Unfortunately, we are not quite done with nomenclature yet. The elementary particles above can form composite particles, too. At one point in time, these composite particles were themselves thought to be elementary, and so many of them were given different names. Luckily for us, we "only" need to remember the following for the purposes of this project:

- Mesons: Υ, B, D, K, π (pion), ρ, a.
- Baryons: p (proton), n (neutron).

Occassionally, some embellishments to the symbols might also be present. Each type of embellishment indicates a different particle, but those particles sharing the same letter are closely related to each other. The following are examples that we will encounter in this project:

- $\Upsilon(4S)$.
- B^+, B^-, B^0, and $\overline{B}^0$.
- D^+, D^-, D^{*0}, $\overline{D}^{*0}$, D^{**}, D_1, D_1', D_2^*, and D_0^*.
- K^+, K^-, K^0, $\overline{K}^0$, K_S^0, and K_L^0.
- π^+, π^-, π^0, ρ^+, ρ^-, ρ^0, and a_1.
- e^+, e^-, μ^+, μ^-, τ^+, τ^-, ν_e, $\overline{\nu}_e$, ν_μ, $\overline{\nu}_\mu$, ν_τ, and $\overline{\nu}_\tau$.

Sometimes we might use the shorthand $D^{(*)}$ to mean that a given expression is valid when either D or D^* is substituted. Similarly, $e^\pm$ can mean that the expression is valid for either e^+ or e^-. When such shorthand is used, the meaning should be evident from the context.

Effectively, all composite particles decay to more elementary constituents after some amount of time. The decayed to particles can, of course, themselves be composite. The decay process continues until all final products are stable. In this project, we consider the following set of *final state particles* to be stable:

$$e^\pm,\ \mu^\pm,\ \pi^\pm,\ K^\pm,\ K_L,\ p,\ n,\ \gamma, \text{ and } \nu.$$

Each composite particle can also decay in several different ways[2]. The *mode* in which a particle decays is a probabilistic process, with some modes being decayed to more frequently than others. It is useful to represent the set of possible decays of a given particle using the following chart[3]. The one shown below represents a small set of possible decays of an Υ particle that we are interested in for this project:

$$\begin{aligned}
\Upsilon &\to BB \\
B &\to D\tau\nu|De\nu|DK|Kee \\
D &\to K\pi|K\pi\gamma \\
\tau &\to e\nu
\end{aligned}$$

Based on this chart, we can "derive" the following example sequence of Υ decays:

$$\Upsilon \Rightarrow BB \Rightarrow D\tau\nu Kee \Rightarrow K\pi\gamma e\nu\nu Kee \tag{1.1}$$

The probability in which a particle decays into a certain mode is often referred to as the mode's *branching fraction*. This is often denoted as $\mathcal{B}(.)$; for instance, $\mathcal{B}(B^0 \to D^- e^+ \nu_e) = 0.0219 \pm 0.0012$.

[2] These are sometimes called the decay *modes* or *channels* of a particle.

[3] This is essentially a *probabilistic context free grammar*. The start symbol is the starting particle, the rules are the decay modes, the non-terminal symbols are composite particles, and the terminal symbols are the final state particles

1.1.3 Particle accelerators, detectors, and their simulations

Data from which physicists use to test the Standard Model are often obtained by operating a particle accelerator. In our case, the accelerator was designed to repeatedly produce the Υ[4] particle by colliding e^+ and e^- particles against each other. The Υ particle then decays naturally into a set of final state particles, which then fly off and strike the detector elements. Such particle *hits* get converted to electrical pulses by the various components of the detector, and are eventually processed and stored into a database.

The data that we receive from operating such an experimental setup is organized as a set of records. By and large, a single record corresponds to a single collision. The contents of the record consists of the hits that were registered, plus any post processing information that were derived from it.

As is the case for many scientific and engineering endeavors, it is often not practical, and sometimes even impossible to obtain the data sample that is required to carry out the statistical inference task of interest. When this is the case, usually after having exhausted all other options, we opt for using a computer to simulate the data sample of interest. The interpretation of our results, are therefore, more nuanced. Throughout this document, we take special care to discuss these finer points whenever they are relevant. Nevertheless, our computer simulated accelerator data is surprisingly accurate in many cases, and serves as a testament to the physicist's understanding of nature and their instruments.

1.2 Project Motivation

The physical quantity we measure is the following ratio of branching fractions:

$$\mathcal{R}(D^{(*)}) = \frac{\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)}{\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)}, \tag{1.2}$$

where ℓ is either e or μ. Strictly speaking, we have defined two $\mathcal{R}(D^{(*)})$ quantities that correspond to the different lepton flavors of ℓ. However, the difference between them is negligible for the purposes of this analysis and we simply refer to the single quantity, $\mathcal{R}(D^{(*)})$.

The initial motivation for performing this analysis is due to *BABAR*'s 2012 result[1]. At the time, this result measured a 3.4σ deviation of $\mathcal{R}(D^{(*)})$ from those predicted in the Standard Model (SM). Since then, many others[2][3][4][5] have contributed their measurements towards a world average. As of summer 2017, the average performed by the Heavy Flavor Averaging Group (HFAG) finds a deviation at 4.1σ relative to the Standard Model calculations performed in [6] and [7]. A visual and tabular summary of these results are shown in figure 1.1 and table 1.1, repectively.

The goal of our analysis is to provide additional evidence, either for or against, the Standard Model.

Perhaps it is worth at least one paragraph to muse about the possibilities for new laws of nature, if not for the purpose of boosting morale and motivation, had the Standard Model been rejected in this particular way. One idea that has been popular, though not as much so recently compared to a few years ago, is that there could be additional Higgs bosons lurking around to be discovered. This possibility was attractive because it is often a required ingredient for some of the following major areas of interest:

- Supersymmetry: from a particle content point of view, the presence of such a symmetry law will necessitate every elementary particle to have a corresponding "super" partner. Of course, this idea was not introduced for the purpose of making the Standard Model more complicated than it already is, but rather, it was introduced as a possible solution to resolve several of its theoretical deficiencies.

- Quantum gravity: the desire for supersymmetry is also strengthened by the desire to build viable string theories. These theories are one of the most promising avenues towards harmonizing relativity with quantum mechanics, the two major pillars from which all contemporary physics are built.

The list above barely scratches the surface of what they are about and of what is possible. I highly encourage the interested reader to find more information elsewhere.

[4] Actually, $\Upsilon(4S)$. But we try to minimize the nomenclature in this chapter.

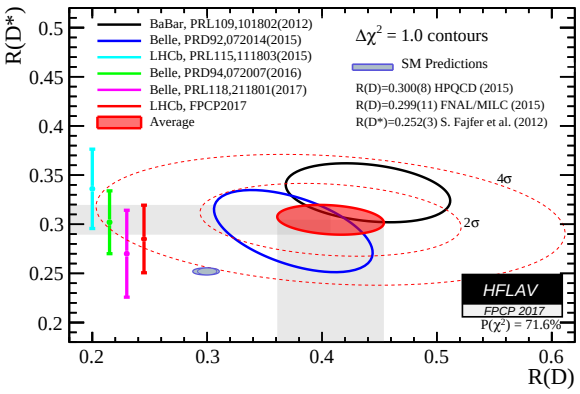


Figure 1.1: Visual summary of recent results. Image courtesy of HFAG.

	$\mathcal{R}(D)$	$\mathcal{R}(D^*)$
BABAR 2013 [1]	$0.440 \pm 0.058 \pm 0.042$	$0.332 \pm 0.024 \pm 0.018$
Belle 2015 [2]	$0.375 \pm 0.064 \pm 0.026$	$0.293 \pm 0.038 \pm 0.015$
LHCb 2015 [4]	$0.336 \pm 0.027 \pm 0.03$	-
Belle 2016 [3]	$0.302 \pm 0.03 \pm 0.011$	-
Belle 2017 [5]	$0.270 \pm 0.035 \pm 0.028$	-
LHCb 2017 [3]	$0.285 \pm 0.019 \pm 0.029$	-
Average (HFAG Summer 2017)	$0.407 \pm 0.039 \pm 0.024$	$0.304 \pm 0.013 \pm 0.007$
Standard Model	0.300 ± 0.008[6]	0.252 ± 0.003[7]

Table 1.1: Tabular summary of recent results. $\mathcal{R}(D)$ and $\mathcal{R}(D^*)$ exceed the SM predictions by 2.3σ and 3.4σ, respectively. Note that we do not list the correlations between the two quantities for each listing, though they are accounted for in the inference procedure. The final combination of the measurements conclude a p-value of 4.13×10^{-5} relative to the Standard Model, which corresponds to a 4.1σ deviation.

Chapter 2

Phenomenology

This chapter discusses the basic phenomenology for $B \to D^{(*)}\ell\nu_\ell$ decays. The material presented here is largely taken from [8], [6], [7], and [9]. We will be skipping over many details that are either technically tricky and/or not particularly illuminating; however, the interested reader can find them in the references given.

Even though the eventual goal of this discussion is to compute the Standard Model predictions for $\mathcal{R}(D^{(*)})$, the main purpose is actually to understand the origins of the theoretical uncertainties. It will turn out that some of these uncertainties are closely related to the systematic uncertainties that enter into our final result, and we pay special attention to point these out when they arise.

This chapter contains technical information intended for the domain expert. Others may safely skip this material.

2.1 General comments

We copy the definition of $\mathcal{R}(D^{(*)})$ below for convenience:

$$\mathcal{R}(D^{(*)}) = \frac{\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)}{\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)}, \tag{2.1}$$

To compute its value using the Standard Model, it suffices to compute the branching fraction $\mathcal{B}(\overline{B} \to M\ell^-\bar{\nu}_\ell)$, where M refers to either a D or a D^* and where ℓ can refer to any of e, μ, or τ. To this end, we must analyze the decay that is shown schematically in figure 2.1.

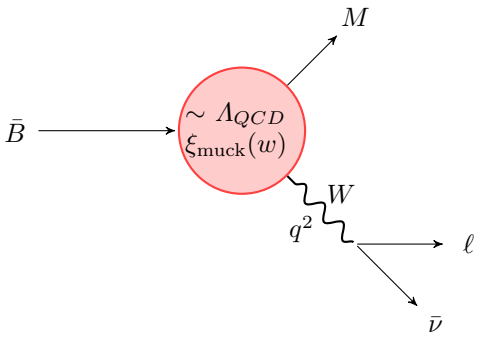


Figure 2.1: B decay schematic.

Before moving on with the calculations, we make some general comments about figure 2.1. The amplitude of $B \to M\ell\nu$ factorizes into two contributions. The first is a well understood electroweak process that corresponds to the lepton production from the virtual W. The second contribution comes from having the B meson convert into a M (D or D^*) meson, which is a process subject to QCD contributions. While much of the QCD effects are well controlled under the heavy quark symmmetry, there are still remaining long distance effects that are encapsulated in the so-called Isgur-Wise function, $\xi_{\text{muck}}(w)$. To understanding the meaning of this function, we define the following kinematic variables:

$$q = p_B - p_M \tag{2.2}$$

and

$$v = p_B/m_B, \qquad v' = p_M/m_B, \qquad w = v \cdot v' = \frac{m_B^2 + m_M^2 - q^2}{2m_B m_M} \tag{2.3}$$

q is simply the momentum that is carried away by the leptons, while v and v' are the 4-velocity of the B and M mesons. When the B meson decays, the color neutral leptonic system carries away momentum, causing the leftover meson to recoil. The recoil induces the meson system to change its velocity from v to v'. The surrounding light quarks and gluon must then reassemble themselves to form the M meson, which may or may not happen successfully. The Isgur-Wise function $\xi_{\text{muck}}(w)$ is the amplitude for such a successful reassembly; the larger the recoil, the less likely that the meson reassembles into the desired target. Such reassemblies proceed through strong interactions at energy scales of the order Λ_{QCD}, and are therefore not under theoretical control. Despite remarkable efforts in constraining the form of ξ, its uncertainty still remains the primary bottleneck for making precise theoretical predictions for $\mathcal{R}(D^{(*)})$.

2.2 Scattering amplitudes

We begin our calculations by writing down the relevant interaction terms in the Standard Model. These are described by the following leading order effective Lagrangian:

$$\mathcal{L}_{\text{eff}} = -\frac{G_F}{\sqrt{2}} V_{cb} J^{\dagger\mu}_{\nu\ell} J_{cb\mu} + \text{h.c.} \tag{2.4}$$

where

$$J^{\mu}_{\nu\ell} \coloneqq \bar{\psi}_\nu \gamma^\mu (1-\gamma^5)\psi_\ell \tag{2.5}$$

$$J^{\mu}_{cb} \coloneqq \bar{\psi}_c \gamma^\mu (1-\gamma^5)\psi_b \tag{2.6}$$

This is actually the familiar 4-point Fermi interaction theory, where the W boson has been formally integrated out. The error we make by focusing only on this term is on the order of $m_b m_\ell / M_W^2$, which is at worst $\sim 10^{-3}$.

We use λ_M, λ_ℓ, and λ_W to denote the helicities for the M, ℓ, and the virtual W. Note that λ_M can take on the values ± 1 and 0 when $M = D^*$, and only 0 when $M = D$. To distinguish the helicity zero states, we use $\lambda_M = s$ to denote the case when $M = D$. This convention also applies to λ_W.

We can express the metric tensor as

$$-g^{\mu\nu} = \sum_{\lambda_W} \eta_{\lambda_W} \epsilon^\mu(\lambda_W) \epsilon^{*\nu}(\lambda_W) \tag{2.7}$$

where we have defined η and the conventions for the polarization tensor of the virtual W as

$$\eta_\pm = \eta_0 = -\eta_s = 1 \tag{2.8}$$

$$\epsilon(q,\pm)^\mu = \mp\frac{1}{\sqrt{2}}(0,1,\mp i,0) \tag{2.9}$$

$$\epsilon(q,0)^\mu = \mp\frac{1}{\sqrt{q^2}}(p_M,0,0,-q^0) \tag{2.10}$$

$$\epsilon(q,s)^\mu = \mp\frac{1}{\sqrt{q^2}} q^\mu \tag{2.11}$$

and identified $\epsilon(\lambda_W)$ as shorthand for $\epsilon(q, \lambda_W)$.

The scattering amplitude for the decay process of figure 2.1 can now be written as

$$\mathcal{M}^{\lambda_\ell}_{\lambda_M}(q^2, \cos\theta_\ell) = \frac{G_F}{\sqrt{2}} V_{cb} \sum_{\lambda_W} \eta_{\lambda_W} L^{\lambda_\ell}_{\lambda_W} H^{\lambda_M}_{\lambda_W} \tag{2.12}$$

where θ_ℓ is the decay angle of the lepton in the W rest frame, and

$$H^{\lambda_M}_{\lambda_W} = \epsilon^*_\mu(\lambda_W) \langle M(p_M, \lambda_M) | J^\mu_{cb} | \overline{B}(p_B) \rangle \tag{2.13}$$

$$L^{\lambda_\ell}_{\lambda_W} = \epsilon_\mu(\lambda_W) \langle \ell(p_\ell, \lambda_\ell) | J^{\dagger\mu}_{\ell\nu} | 0 \rangle \tag{2.14}$$

describe the $\overline{B} \to MW^*$ and $W^* \to \ell\bar{\nu}$ decays, repectively.

The evaluation of the leptonic amplitudes is, in principle, straightforward. Following the coordinate and phase conventions given in [8], we get

$$L^-_\pm(q^2, \cos\theta_\ell) = \sqrt{2q^2}\sqrt{1 - m_\ell^2/q^2}(1 \pm \cos\theta_\ell) \tag{2.15}$$

$$L^-_0(q^2, \cos\theta_\ell) = -2\sqrt{q^2}\sqrt{1 - m_\ell^2/q^2}\sin\theta_\ell \tag{2.16}$$

$$L^-_s(q^2, \cos\theta_\ell) = 0 \tag{2.17}$$

and

$$L^+_\pm(q^2, \cos\theta_\ell) = \pm\sqrt{2} m_\ell \sqrt{1 - m_\ell^2/q^2}\sin\theta_\ell \tag{2.18}$$

$$L^+_0(q^2, \cos\theta_\ell) = 2m_\ell \sqrt{1 - m_\ell^2/q^2}\cos\theta_\ell \tag{2.19}$$

$$L^+_s(q^2, \cos\theta_\ell) = -2m_\ell \sqrt{1 - m_\ell^2/q^2} \tag{2.20}$$

Note that the L^+ amplitudes are helicity supressed by the factor m_ℓ, and L^-_0 vanishes by angular momentum conservation.

The hadronic amplitudes are written in terms of a set of *form factors*. These are precisely the functions that include the long distance QCD effects involved during hadronization.

2.2.1 Hadronic amplitudes for $B \to D\ell\nu$

The convention for expressing the $B \to D\ell\nu$ transition is in terms of two form factors $h_\pm(w)$:

$$\langle D(v') | \bar{c}\gamma^\mu b | \overline{B}(v) \rangle = \sqrt{m_B m_D}(h_+(w)(v + v')^\mu + h_-(w)(v - v')^\mu) \tag{2.21}$$

This gives the following expressions for the hadronic amplitudes:

$$H^s_\pm(q^2) = 0 \tag{2.22}$$

$$H^s_0(q^2) = \sqrt{m_B m_D}\frac{1 + r}{\sqrt{1 - 2rw + r^2}}\sqrt{w^2 - 1}V_1(w) \tag{2.23}$$

$$H^s_s(q^2) = \sqrt{m_B m_D}\frac{1 - r}{\sqrt{1 - 2rw + r^2}}(w + 1)S_1(w) \tag{2.24}$$

where $r = m_D/m_B$ and

$$V_1(w) = h_+(w) - \frac{1 - r}{1 + r}h_-(w) \tag{2.25}$$

$$S_1(w) = h_+(w) - \frac{1 + r}{1 - r}\frac{w - 1}{w + 1}h_-(w) \tag{2.26}$$

Note that we liberally interchange between using q^2 and w, and also between using p and v. These are equivalent ways of expression, but one set of conventions is sometimes more convenient than another.

A popular way of parametrizing $V_1(w)$ and $S_1(w)$ is to use dispersion relation constraints in a QCD sum rule [10]. This gives the following expressions:

$$V_1(w) = V_1(1)(1 - 8\rho_1^2 z + (51\rho_1^2 - 10)z^2 - (252\rho_1^2 - 84)z^3) \tag{2.27}$$

$$S_1(w) = (1 + \Delta(w))V_1(w) \tag{2.28}$$

where

$$\Delta(w) = -0.019 + 0.041(w-1) - 0.015(w-1)^2 \tag{2.29}$$

$$z = \frac{\sqrt{w+1} - \sqrt{2}}{\sqrt{w+1} + \sqrt{2}} \tag{2.30}$$

and the slope parameter $\rho_1 = 1.18 \pm 0.04 \pm 0.04$ is determine using experimental data, which we obtain from HFAG [11]. $V_1(1)$ is the term that encapsulate the degree for which heavy quark symmetry holds; in fact, $V(w)$ *is* the Isgur-Wise function disguised, and $V(1) = 1$ if the symmetry is exact. While we can use heavy quark effective theory (HQET) to systematically determine any symmetry breaking contributions to $V(1)$, its exact value will not matter in predictions for $\mathcal{R}(D)$ since it will cancel in the ratio.

2.2.2 Hadronic amplitudes for $B \to D^* \ell \nu$

The convention for expressing the $B \to D^* \ell \nu$ transition is in terms of four form factors $V(q^2)$, $A_0(q^2)$, $A_1(q^2)$, and $A_2(q^2)$:

$$\langle D^*(p_{D^*}, \lambda_{D^*}) | \bar{c}\gamma^\mu b | \overline{B}(p_B) \rangle = \frac{2iV(q^2)}{m_B + m_{D^*}} \epsilon_{\mu\nu\alpha\beta} \epsilon^{*\nu} p_B^\alpha p_{D^*}^\beta \tag{2.31}$$

$$\langle D^*(p_{D^*}, \lambda_{D^*}) | \bar{c}\gamma^\mu \gamma^5 b | \overline{B}(p_B) \rangle = 2m_{D^*} A_0(q^2) \frac{\epsilon^* \cdot q}{q^2} q_\mu + (m_B + m_{D^*}) A_1(q^2) \left(\epsilon_\mu^* - \frac{\epsilon^* \cdot q}{q^2} q_\mu \right) \tag{2.32}$$

$$- A_2(q^2) \frac{\epsilon^* \cdot q}{m_B + m_{D^*}} \left((p_B + p_{D^*})_\mu - \frac{m_B^2 - m_{D^*}^2}{q^2} q_\mu \right) \tag{2.33}$$

where the polarization tensor ϵ is shorthand for $\epsilon(\lambda_{D^*})$.

The hadronic amplitudes are then expressed in terms of these form factors as

$$H_\pm^\pm(q^2) = (m_B + m_{D^*}) A_1(q^2) \mp \frac{2m_B}{m_B + m_{D^*}} |\vec{p}_{D^*}| V(q^2) \tag{2.34}$$

$$H_0^0(q^2) = \frac{1}{2m_{D^*}\sqrt{q^2}} ((m_B^2 - m_{D^*}^2 - q^2)(m_B + m_{D^*}) A_1(q^2) - \frac{4m_B^2 |\vec{p}_{D^*}|^2}{m_B + m_{D^*}} A_2(q^2)) \tag{2.35}$$

$$H_s^0 = \frac{2m_B |\vec{p}_{D^*}|}{\sqrt{q^2}} A_0(q^2) \tag{2.36}$$

The most popular way to parametrize the form factors are those of reference [10]. They begin by introducing a new set of form factors $h_{A_1}(w)$, $R_0(w)$, $R_1(w)$, and $R_2(w)$. They are related to the conventional

form factors as follows:

$$h_{A_1}(w) \coloneqq A_1(q^2)\frac{1}{R_{D^*}}\frac{2}{w+1} \tag{2.37}$$

$$A_0(q^2) = \frac{R_0(w)}{R_{D^*}}h_{A_1}(w) \tag{2.38}$$

$$A_2(q^2) = \frac{R_2(w)}{R_{D^*}}h_{A_1}(w) \tag{2.39}$$

$$V(q^2) = \frac{R_1(w)}{R_{D^*}}h_{A_1}(w) \tag{2.40}$$

$$\tag{2.41}$$

where $R_{D^*} = 2\sqrt{m_B m_{D^*}}/(m_B + m_{D^*})$. They then apply analyticity constraints to deduce the following parametrization of the form factors:

$$h_{A_1}(w) = h_{A_1}(1)(1 - 8\rho^2 z + (53\rho^2 - 15)z^2 - (231\rho^2 - 91)z^3) \tag{2.42}$$

$$R_1(w) = R_1(1) - 0.12(w-1) + 0.05(w-1)^2 \tag{2.43}$$

$$R_2(w) = R_2(1) + 0.11(w-1) - 0.06(w-1)^2 \tag{2.44}$$

$$R_0(w) = R_0(1) - 0.11(w-1) + 0.01(w-1)^2 \tag{2.45}$$

where $h_{A_1}(1)$, $R_1(1)$, $R_2(1)$, and ρ are taken from HFAG. Similar to $V_1(1)$, $R_0(1)$ is also a term that encapsulates the degree to which heavy quark symmetry holds. Unlike $V_1(1)$, however, it neither cancels in the ratio $\mathcal{R}(D^*)$, nor is it presently amenable to experimental determination. Its value is thus determined systematically from HQET, which turns out to be 1.14 (see Appendix B of [7] for details).

2.3 Decay rates

The differential decay rate for $B \to D^{(*)}\ell\nu$ is then expressed in terms of the scattering amplitude as follows

$$d\Gamma = \frac{1}{2m_B}\sum_{\lambda_\ell \lambda_M} |\mathcal{M}^{\lambda_\ell}_{\lambda_M}|^2 d\Phi_3 \tag{2.46}$$

where

$$d\Phi_3 = \frac{\sqrt{Q_+ Q_-}}{256\pi^3 m_B^2}\left(1 - \frac{m_\ell^2}{q^2}\right) dq^2 d\cos\theta_\ell \tag{2.47}$$

and $Q_\pm = (m_B \pm m_M)^2 - q^2$.

Substituting the expressions for the scattering amplitudes, we get

$$\frac{d^2\Gamma}{dq^2 d\cos\theta} = \frac{G_F^2 |V_{cb}|^2 |\vec{p}_M| q^2}{256\pi^3 m_B^2}\left(1 - \frac{m_\ell^2}{q^2}\right)^2 \times \tag{2.48}$$

$$((1-\cos\theta)^2|H_+^+|^2 + (1+\cos\theta)^2|H_-^-|^2 + 2\sin^2\theta|H_0^0|^2 + \tag{2.49}$$

$$\frac{m_\ell^2}{q^2}\left(\sin^2\theta(|H_+^+|^2 + |H_-^-|^2) + 2|H_s^0 - H_0^0\cos\theta|^2\right)) \tag{2.50}$$

Performing the integration over $d\cos\theta$, we obtain

$$\frac{d\Gamma}{dq^2} = \frac{G_F^2 |V_{cb}|^2 |\vec{p}_M| q^2}{96\pi^3 m_B^2}\left(1 - \frac{m_\ell^2}{q^2}\right)^2 \left((|H_+^+|^2 + |H_-^-|^2 + |H_0^0|^2)\left(1 + \frac{m^2\ell}{2q^2}\right) + \frac{3}{2}\frac{m_\ell^2}{q^2}|H_s^0|^2\right) \tag{2.51}$$

The differential decay rate formulae above are useful later on when we want to change form factor models by "re-weighting". For a full discussion on this topic, see reference [9].

Finally, integrating over dq^2 and substituting the appropriate masses for $\ell = e, \tau$ gives the Standard Model predictions for $\mathcal{R}(D^{(*)})$:

$$\mathcal{R}(D)_{SM} = 0.302 \pm 0.015 \tag{2.52}$$

$$\mathcal{R}(D^*)_{SM} = 0.252 \pm 0.003 \tag{2.53}$$

The origins of these theoretical uncertainties is interesting. As one might have expected, the uncertainty in the slope parameter ρ for the Isgur-Wise functions is indeed a major contributor, and is the main uncertainty that dominates for $\mathcal{R}(D)$. For $\mathcal{R}(D^*)$, however, the parameter $R_0(1)$ is also a major contributor; the authors of [7] claim that this is the main effect for the uncertainty in $\mathcal{R}(D^*)$, and can be improved by more precise lattice QCD computations. It is also interesting to note that ρ is determined entirely from a single analysis by Belle [12], so any mistakes and unaccounted systematics could also skew the results of these "theory" predictions.

Chapter 3

PEP-II and *BABAR*

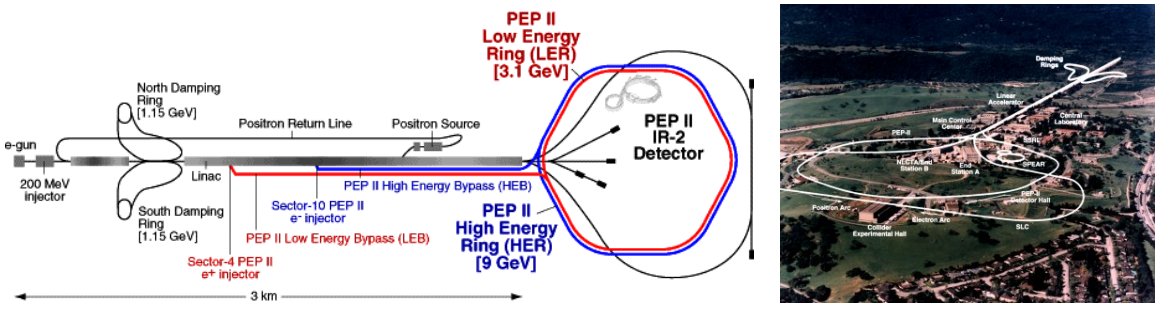
In chapter 1, we discussed an analysts' perspective of the data. While this level of abstraction is extremely useful and perhaps sufficient for the purposes of our project, the actual data generation and collection is *far* more complicated. If our abstraction were taken literally, it would be at least as outrageous as saying that a computer is "just" a CPU and memory, and a car is "just" a peddle and a steering wheel.

For one, the $\Upsilon(4S)$ particles don't just appear from thin air. It is produced by accelerating electrons and positrons and colliding them under extreme, yet controlled conditions. The machine that generates these collisions is called *PEP-II*.

The detection of the $\Upsilon(4S)$ and its decay products is not automatic, either. Sure, the detector is like a camera, but what components are necessary to be installed and what are their design requirements? The *BABAR* detector is what is used to actually collect the data. It has several major subsystems, all of which are designed and optimized to collect data of the highest quality.

This chapter will only outline the key aspects that underlies the design and operation of PEP-II and *BABAR*. For a full discussion, see [13].

3.1 The PEP-II accelerator



(a) Schematic.

(b) Annotated aerial view.

Figure 3.1: PEP-II

The main component of PEP-II consists of two storage rings. The high energy storage ring (HER) delivers 9 GeV electrons while the low energy storage ring (LER) delivers 3.1 GeV electron. These settings were decided such that the center of mass energy would be 10.58 GeV, corresponding to the mass of the $\Upsilon(4S)$.

You might wonder why the beam energies are asymmetric. The reason is that one of the primary goals of *BABAR* was to make a precise measurement of $\sin 2\beta$, which to a good approximation, quantifies the amount of CP violation present in $B^0\overline{B}^0$ mixing. In order to do so, there had to be sufficient position separation

between the two B^0 meson decays such that the detector could resolve their oscillation structure. The asymmetric beam energy implements just that by causing a sufficiently large boost for the center of mass frame. This leads to large B meson velocities in the lab frame, enough that the decay distances could be resolved.

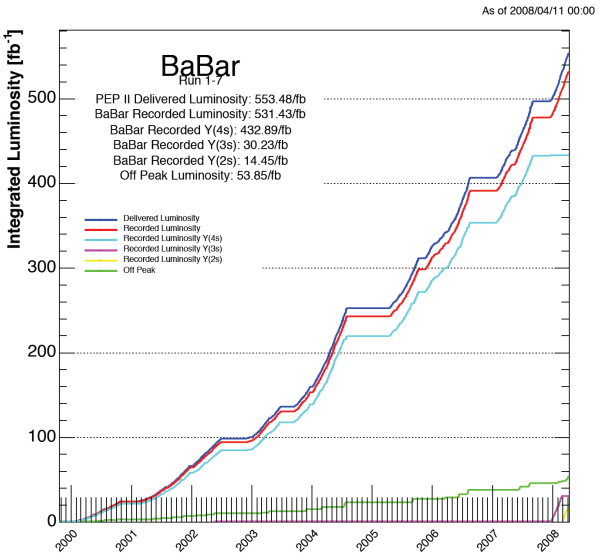


Figure 3.2: Integrated luminosity over time.

PEP-II was designed to operate at an instantaneous luminosity of 3×10^{33} cm^{-2}s^{-1}. However, the instrument was designed so well that it was able to operate at 12×10^{33} cm^{-2}s^{-1}, which is 4 times the design. Between the years of its operation (1999-2008), it delivered an integrated luminosity of 553.48 fb^{-1}, of which 432.89 fb^{-1} were recorded at the $\Upsilon(4S)$ resonance ("on-peak"), and 53.85 fb^{-1} was recorded in the continuum ("off-peak"). Our project uses all of the off-peak data as well as 429.89 fb^{-1} of the on-peak data. This corresponds to about 471×10^6 $B\overline{B}$ pairs. Figure 3.2 shows the integrated luminosity over the years of operation.

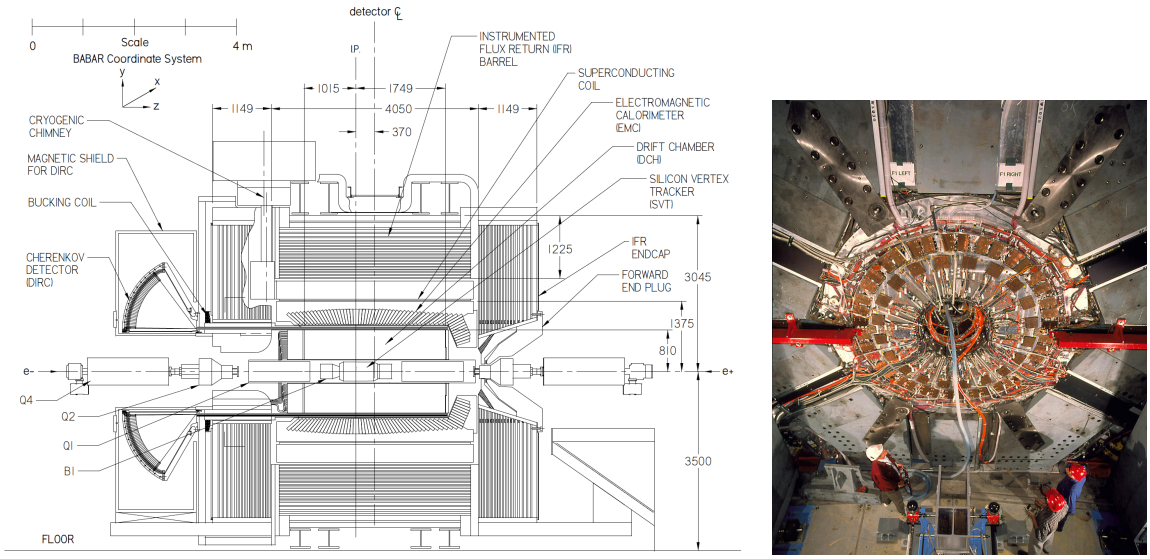
3.2 The *BABAR* detector

Many particle detectors, including *BABAR*, are interested in directly detecting the decay products caused by the collision. Decay products can be grouped into two types: those that are stable (final state particles) and those that are not (intermediate particles). Roughly speaking, the word "stable" refers to whether the particle would decay before it reaches the detector, and so these are the only particles that are directly detectable. The detection of unstable particles are done indirectly, by forming combinations of stable ones and checkig whether the result satisfies characteristics of the target. Since all inference about the collision relies on the detection of stable particles, it is no surprise that the detector is optimized for it.

Detecting a particle usually means that we measure its position of origination[1], its energy, and its momenta. These are measured by carefully inspecting its flight trajectory, which are observed directly from its interaction with the various detector elements and pixels. Sometimes it is even possible to decide on the species of a stable particle by observing the manner in which its trajectory has interacted with the detector.

Figure 3.3 shows the *BABAR* detector. It consists of several layers of subsystems; they are the silicon vertex tracker (SVT), the drift chamber (DCH), the cherenkov detector (DIRC), the electromagnetic calorimeter (EMC), a superconducting coil that provides 1.5 T of solenoidal magnetic field, and the instrumented flux

[1]This is sometimes referred to as the *vertex*. This usage is most common when discussing unstable particles.



(a) Schematic.

(b) In real life.

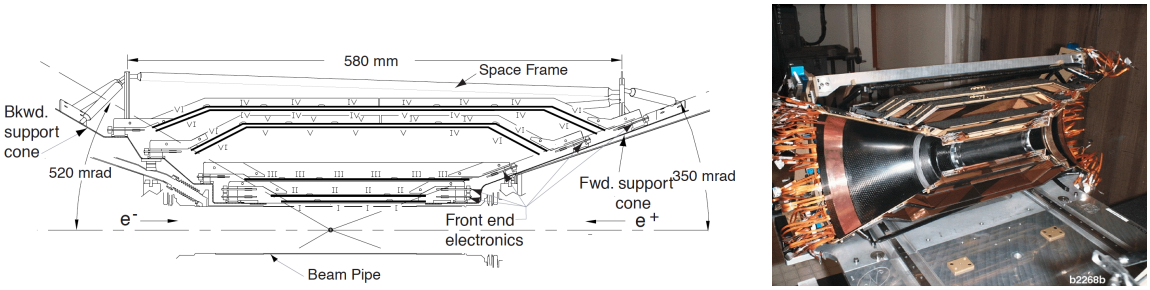
Figure 3.3: The *BABAR* detector.

return (IFR). We will discuss the functionality of many of these systems in the subsections below and its primary role in the detection process.

The design of *BABAR* optimized on the following features:

1. Large solid angle coverage in the center of mass frame.
2. High vertex and 4-momenta resolution.
3. Particle identification.

3.2.1 Silicon vertex tracker (SVT)



(a) Schematic.

(b) In real life.

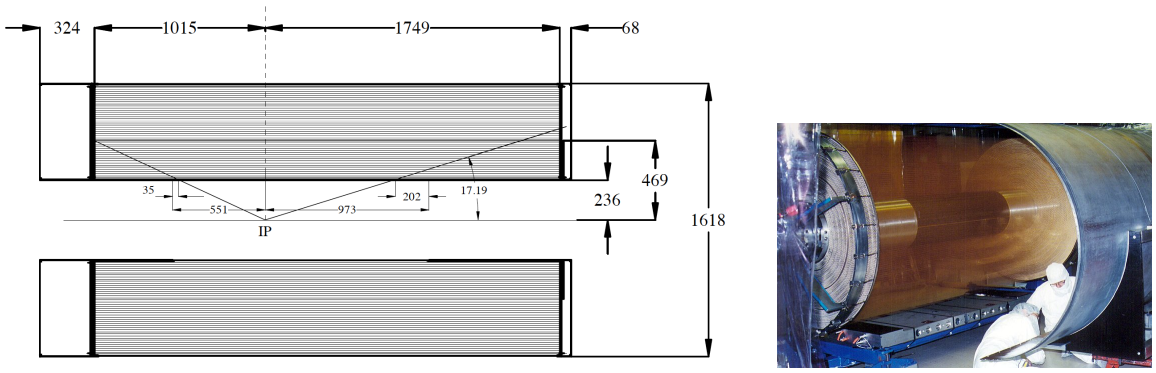
Figure 3.4: Silicon vertex tracker.

The silicon vertex tracker is one of two components in *BABAR* responsible for detecting charged particle trajectories. The idea is that a charged particle could interact with its layers so that the signals could be used to accurately register the sequence of hit positions.

It is organized as 5 layers of double sided silicon strips. The strips on opposite sides of the sensor are orthogonal to each other: the ϕ measuring strip runs parallel to the beam pipe while the other measures the z position.

Despite having seemingly few layers, the virtue of the SVT is that it is as close to the interaction point as possible. The further one moves from the interaction point, the noisier the signal becomes since the particles would have gone through more iterations of scattering. In fact, the SVT is what provides the bulk of the resolution for B and D meson vertices.

3.2.2 Drift chamber (DCH)



(a) Schematic.

(b) In real life.

Figure 3.5: Drift chamber.

The drift chamber is the other component in *BABAR* responsible for detecting charged particle trajectories. It is located further away from the interaction point, but provides many more charged particle hits. Together with hits registered in the SVT, the sequence of all hits are used to fit for the particle flight trajectory. Since the solenoidal magnetic field is also known, the flight trajectory is also sufficient for determining the particle's three momenta.

The DCH consists of 40 layers of small hexagonal cells. The total number of drift cell is 7,104. Each cell consists of 1 tungsten-rhenium sense wire and 6 aluminum field wires. The chamber is about 3m long and filled with a mixture of 80% helium and 8-9.5% isobutane to minimize multiple scattering.

An additional functionality of the DCH is to observe dE/dx of charged particles. This is an additional feature that is input towards particle identification.

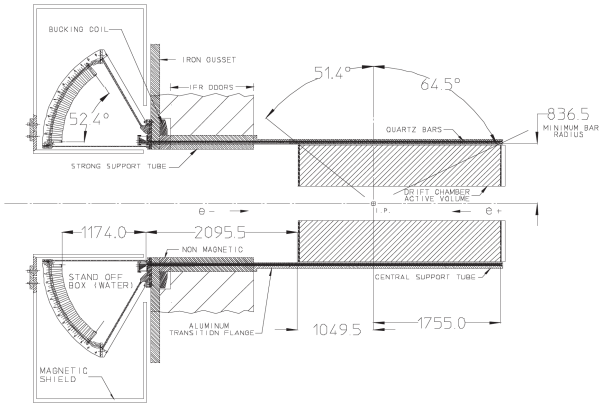
3.2.3 Cherenkov detector (DIRC)

The purpose of the Detector of Internally Reflected Cherenkov light (DIRC) is to observe additional features relevant for charged particle identification. It operates on the principle that when a charge particle travels through a fused silica bar, it emits Cherenkov radiation. The opening cone angle (θ_c) is related to the velocity in which the particle moves through the material.

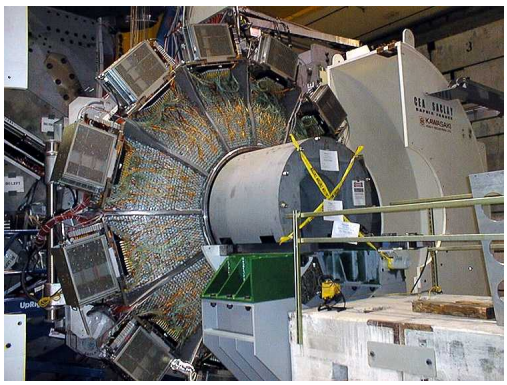
The DIRC consists of thin synthetic fused silica bars with one end equipped with a mirror and the other end connected to a photo multiplier tube (PMT) array viewing a water container.

3.2.4 Electromagnetic calorimeter (EMC)

The EMC consists of a cylindrical barrel and a conical forward endcap; the layout captures about 90% of the total solid angle of the center of mass system. The barrel contains 5,760 thallium doped CsI crystals

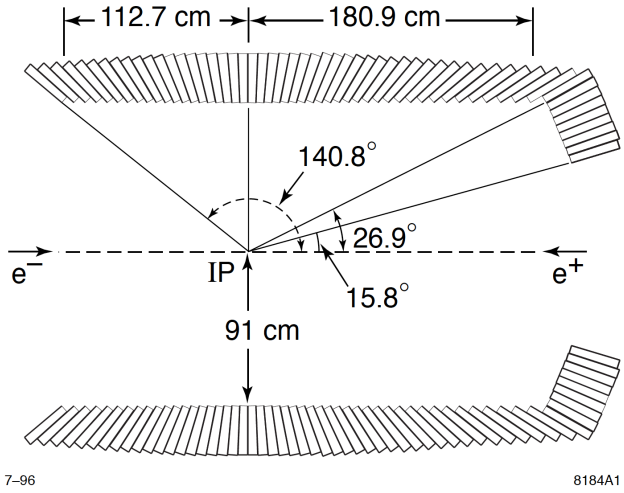


(a) Schematic.



(b) In real life.

Figure 3.6: The DIRC.



(a) Schematic.

(b) In real life.

Figure 3.7: The EMC.

arranged in 48 distinct rings with 120 crystals each. The endcap holds 820 crystals arranged in 8 rings. Each crystal also has 2 PIN diodes attached at the end.

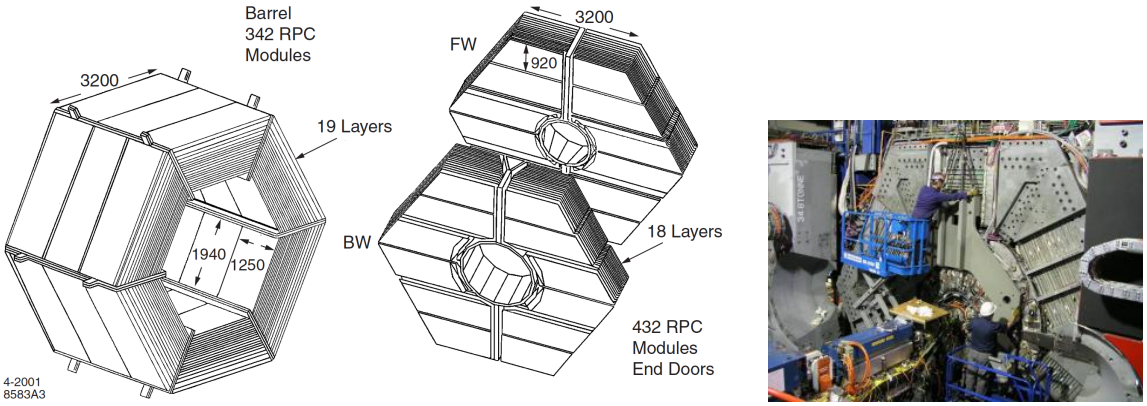
The primary purpose of the EMC is to measure the energy of photons and electrons. When a photon or an electron travels through the crystal, it triggers a cascade of electromagnetic interactions that deposits almost all of its energy to low energy phtons. The light is then reflected inside the crystal and collected with the PIN diodes. The EMC also provides angular resolution for these particles as well. Achieving good photon resolution is also very important for detecting π^0's since it decays to two photons.

3.2.5 Instrumented flux return (IFR)

The IFR serves a dual purpose as being the steel flux return for the magnet as well as being a muon and long lived hadron filter. Its inputs are then used to in particle identification.

During the early phases of the experiment, resistive plate chambers (RPC) were installed between the steel plates. Due to serious aging problems of the RPCs, the IFR underwent major upgrades. The original

chambers in the forward endcap were replaced with RPCs of an improved design, while the barrel chambers were replaced by limited streamer tubes (LST).



(a) Schematic. (b) In real life.

Figure 3.8: The IFR.

Chapter 4

Analysis Strategy

4.1 Overview

The entry point to this analysis are the particle list primitives provided by the *BABAR* software framework. We then proceed in the following steps to obtain the result:

1. Event reconstruction: For each event record, we use the particle list primitives to reconstruct a decay prototype of interest. In particular, we reconstruct all possible decay sequences that proceeds through a selected set of modes that a $\Upsilon(4S)$ could have taken to yield an event record consistent with the particle list primitives.

2. Feature extraction: For each event as well as for each reconstructed $\Upsilon(4S)$ candidate, we compute several features associated with them that are useful for distinguishing between *signal* and *background* (to be defined in section 4.2).

3. Candidate selection: Event reconstruction can yield up to several $\Upsilon(4S)$ candidates for each event. While it is possible to perform the analysis by pooling information over all candidates, we chose to elect a single candidate to represent the event.

4. Signal classification: This stage uses the features extracted from each event to assign a score that characterizes how similar it is to signal or to background.

5. Signal extraction: This steps estimates the number of events that are signal. As will be shown later, the count obtained in this step divided by the efficiencies obtained in the next step will determine the central value of the final result.

6. Efficiency estimation: This steps estimates the signal efficiency; that is, the probability that an event passes all filters, given that it is a signal event.

7. Quantify systematic uncertainties.

8. Construct confidence regions.

In what follows, we begin with some definitions as well as a derivation of the main statistical quantity of interest. We then proceed to describe how the main inference task is factorized and how each subproblem is solved using the steps outlined above.

4.1.1 Terminology

We must digress to discuss terminology that is not necessarily standard across physics analyses.

Particle physicists often refer to an accelerator collision as an "event". This terminology is consistent with the usage commonly encountered in relativity to refer to any single point in spacetime. Unfortunately, the term "event" also has a special meaning in probability theory. In that discipline, it is defined to be any subset of the sample space.

Since our analysis methodology relies heavily on probabilistic analysis, we use the following terminology unless specifically stated otherwise:

Term	Meaning
(collision) outcome	a single accelerator collision
event	subset of the sample space

4.2 Event Categories

Before proceeding, we must define the probabilistic setting from which we are to perform statistical inference.

One can think of any single collision outcome at *BABAR* as a sample from the probability space $(\Omega, \mathcal{U}, \mathbb{P})$, where Ω is the set of all possible outcomes, and $\mathcal{U}$, $\mathbb{P}$ are suitably defined such that all outcomes are equally likely.

We now define the set of *event types* as

$$C = \{D\tau_h,\ D^*\tau_h,\ D^{**}SL,\ Comb,\ Cont\}, \tag{4.1}$$

and a mapping $\mathcal{T} : \Omega \to C$ according to the decision tree shown in figure 4.1. In particular, for any $\omega \in \Omega$, $\mathcal{T}(\omega)$ is defined to be the leaf label that the decision tree assigns to ω.

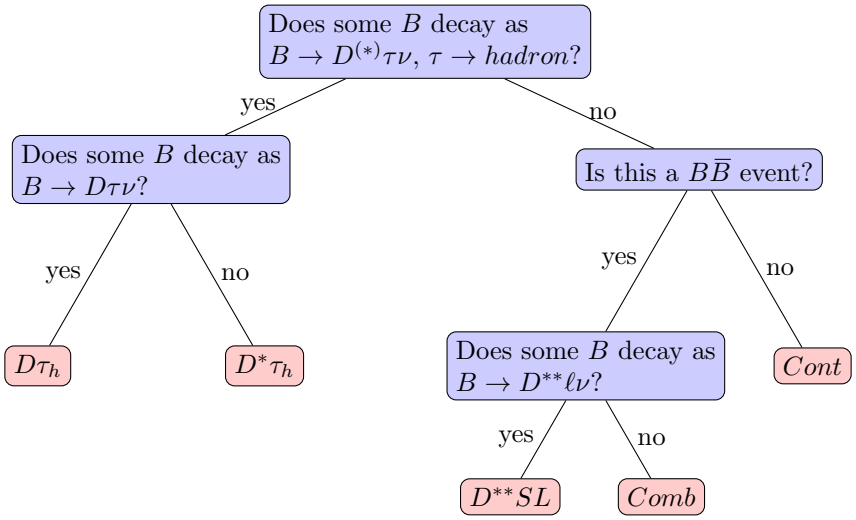


Figure 4.1: The leaves of this decision tree define a partitioning of *BABAR* events.

Note that $\mathcal{T}$ defines a partition $\{E_i\}_{i \in C}$ over Ω[1].

We can now define the following events and their corresponding aliases:

Event Definition	Alias
$E_{B\overline{B}} = \cup_{i \in C \setminus \{Cont\}} E_i$	$B\overline{B}$
$E_S = E_{D\tau_h} \cup E_{D^*\tau_h}$	*signal*
$E_B = E_S^c$	*background*

We may occassionally add a qualifier to isolate a specific type of signal or background; e.g. $D\tau_h$ signal refers specifically to $E_{D\tau_h}$, and $B\overline{B}$ background refers to $E_{B\overline{B}} \cap E_B = E_{D^{**}SL} \cup E_{Comb}$.

4.3 Estimating $\mathcal{R}(D^{(*)})$

To measure $\mathcal{R}(D^{(*)})$, it suffices to measure $\mathcal{B}(B \to \overline{D}^{(*)}\tau\nu_\tau)$ and divide by $\mathcal{B}(B \to \overline{D}^{(*)}\ell\nu_\ell)$ obtained from other *BABAR* measurements[14][15][16][17] or the world average[18].

[1] That is, $\Omega = \cup_{i \in C} E_i$ and $E_i \cap E_j = \emptyset$ for all $i \neq j$, $i, j \in C$.

Define the following quantities:

$$P = \mathbb{P}\left[E_{D\tau_h}|E_{B\bar{B}}\right] \tag{4.2}$$
$$Q = \mathbb{P}\left[E_{D^*\tau_h}|E_{B\bar{B}}\right] \tag{4.3}$$
$$p = \mathcal{B}(B \to \bar{D}\tau\nu) \times \mathcal{B}(\tau \to \text{hadrons}) \tag{4.4}$$
$$q = \mathcal{B}(B \to \bar{D}^*\tau\nu) \times \mathcal{B}(\tau \to \text{hadrons}) \tag{4.5}$$

By the elementary rules of probability, we have

$$P = \mathbb{P}\left[E_{D\tau_h}|E_{B\bar{B}}\right] \tag{4.6}$$
$$= \mathbb{P}\left[\text{At least 1 } B \text{ decays as } B \to \bar{D}\tau\nu|E_{B\bar{B}}\right] \tag{4.7}$$
$$= p^2 + 2p(1-p) \tag{4.8}$$
$$\Rightarrow p = 1 - \sqrt{1-P}, \tag{4.9}$$

hence

$$\boxed{\mathcal{B}(B \to \bar{D}\tau\nu) = \frac{1-\sqrt{1-P}}{\mathcal{B}(\tau \to \text{hadrons})}} \tag{4.10}$$

Similarly,

$$Q = \mathbb{P}\left[E_{D^*\tau_h}|E_{B\bar{B}}\right] \tag{4.11}$$
$$= \mathbb{P}\left[\text{At least 1 } B \text{ decays as } B \to \bar{D}^*\tau\nu, \text{ but neither decays as } B \to \bar{D}\tau\nu|E_{B\bar{B}}\right] \tag{4.12}$$
$$= q^2 + 2q(1-p-q) \tag{4.13}$$
$$\Rightarrow q = (1-p)\left(1 - \sqrt{1 - \frac{Q}{(1-p)^2}}\right), \tag{4.14}$$

hence

$$\boxed{\mathcal{B}(B \to \bar{D}^*\tau\nu) = \frac{1-p}{\mathcal{B}(\tau \to \text{hadrons})}\left(1 - \sqrt{1 - \frac{Q}{(1-p)^2}}\right)} \tag{4.15}$$

Observe that the problem of estimating $\mathcal{R}(D^{(*)})$ reduces to the problem of estimating $\mathbb{P}\left[E_{D^{(*)}\tau_h}|E_{B\bar{B}}\right]$. We do, however, require the additional quantities $\mathcal{B}(B \to D^{(*)}\ell\nu_\ell)$ and $\mathcal{B}(\tau \to \text{hadrons})$ to complete the measurement. These values are derived from inputs taken from the PDG and HFAG as follows [18][11]:

$$\mathcal{B}(B \to D\ell\nu_\ell) = \mathbb{P}\left[B \to D\ell\nu_\ell|B = B^\pm\right]\mathbb{P}\left[B = B^\pm\right] + \mathbb{P}\left[B \to D\ell\nu_\ell|B = B^0\right]\mathbb{P}\left[B = B^0\right] \tag{4.16}$$
$$= 0.487 \times \mathcal{B}(B^0 \to \bar{D}^0\ell\nu_\ell) + 0.513 \times \mathcal{B}(B^+ \to D^+\ell\nu_\ell) \tag{4.17}$$
$$= (2.22 \pm 0.10)\% \tag{4.18}$$
$$\mathcal{B}(B \to D^*\ell\nu_\ell) = \mathbb{P}\left[B \to D^*\ell\nu_\ell|B = B^\pm\right]\mathbb{P}\left[B = B^\pm\right] + \mathbb{P}\left[B \to D^*\ell\nu_\ell|B = B^0\right]\mathbb{P}\left[B = B^0\right] \tag{4.19}$$
$$= 0.487 \times \mathcal{B}(B^0 \to \bar{D}^{*0}\ell\nu_\ell) + 0.513 \times \mathcal{B}(B^+ \to D^{*+}\ell\nu_\ell) \tag{4.20}$$
$$= (5.13 \pm 0.11)\% \tag{4.21}$$
$$\mathcal{B}(\tau \to \text{hadrons}) = 1 - \mathcal{B}(\tau \to e\bar{\nu}_e\nu_\tau) - \mathcal{B}(\tau \to \mu\bar{\nu}_\mu\nu_\tau) - \mathcal{B}(\tau \to e\bar{\nu}_e\nu_\tau\gamma) - \mathcal{B}(\tau \to \mu\bar{\nu}_\mu\nu_\tau\gamma) \tag{4.22}$$
$$= (63 \pm 0.18)\% \tag{4.23}$$

where we have used the following PDG values:

$$\mathcal{B}(B^0 \to \overline{D}^0 \ell \nu_\ell) = (2.13 \pm 0.095)\%$$
$$\mathcal{B}(B^+ \to D^+ \ell \nu_\ell) = (2.30 \pm 0.10)\%$$
$$\mathcal{B}(B^0 \to \overline{D}^{*0} \ell \nu_\ell) = (4.93 \pm 0.11)\%$$
$$\mathcal{B}(B^+ \to D^{*+} \ell \nu_\ell) = (5.31 \pm 0.12)\%$$
$$\mathcal{B}(\tau^- \to e \bar{\nu}_e \nu_\tau) = (17.83 \pm 0.04)\%$$
$$\mathcal{B}(\tau^- \to e \bar{\nu}_e \nu_\tau \gamma) = (1.75 \pm 0.18)\%$$
$$\mathcal{B}(\tau^- \to \mu \bar{\nu}_\mu \nu_\tau) = (17.41 \pm 0.04)\%$$
$$\mathcal{B}(\tau^- \to \mu \bar{\nu}_\mu \nu_\tau) = (0.0036 \pm 0.0004)\%$$

4.4 Estimating $\mathbb{P}\left[E_{D^{(*)}\tau_h} | E_{B\overline{B}}\right]$

For reasons described in section 4.5, we will filter events by a set of criteria. Let $U \subseteq \Omega$ be the set of outcomes that will pass this filter and define

$$U_i = U \cap E_i, \qquad \text{for all } i \in C. \tag{4.24}$$

Define the *efficiency* and *sample proportion* for $i \in C$ to be, respectively,

$$\epsilon_i \coloneqq \mathbb{P}\left[U_i | E_i\right] = \frac{|U_i|}{|E_i|} \tag{4.25}$$
$$p_i \coloneqq \mathbb{P}\left[U_i | U\right] = \frac{|U_i|}{|U|}. \tag{4.26}$$

We can now express $\mathbb{P}\left[E_i | E_{B\overline{B}}\right]$ for $i = D\tau_h$, $D^*\tau_h$ as follows:

$$\mathbb{P}\left[E_i | E_{B\overline{B}}\right] = \frac{|E_i|}{|E_{B\overline{B}}|} = \frac{p_i |U|}{\epsilon_i |E_{B\overline{B}}|}. \tag{4.27}$$

The approach we take towards estimating $\mathbb{P}\left[E_{D^{(*)}\tau_h} | E_{B\overline{B}}\right]$ is to simply estimate each term in equation 4.27.

We can estimate $|U|/|E_{B\overline{B}}|$ as follows. Let $\mathcal{D}$ be the data sample that we have collected after all filtering criteria have been applied. That is, each datum corresponds to a single outcome in event U. Use N to denote the size of this sample. *BABAR* has also provided an estimate of the number of $B\overline{B}$ events that it has generated in total. Use $N_{B\overline{B}}$ to denote this estimate. Given this information, we simply estimate $\mathbb{P}\left[E_{D^*\tau_h} | E_{B\overline{B}}\right]$ as follows:

$$\boxed{\hat{\mathbb{P}}\left[E_{D^*\tau_h} | E_{B\overline{B}}\right] = \frac{\hat{p}_{D^{(*)}\tau_h} N}{\hat{\epsilon}_{D^{(*)}\tau_h} N_{B\overline{B}}}} \tag{4.28}$$

where $\hat{p}_{D^{(*)}\tau_h}$ and $\hat{\epsilon}_{D^{(*)}\tau_h}$ are the estimated quantities in sections 4.6 and 4.7.

4.5 Data filtering and feature extraction

Our exposition in this chapter has, thus far, not depended on the specific details of the data filter U and what observations were recorded for each outcome. This section will define the data filtering criteria, i.e. the rules that decide whether an outcome belongs to $U \subseteq \Omega$, and discuss how observations are made as well as how they are modelled in our probabilistic framework.

4.5.1 Data filtering

While it is possible to use the dataset as is without applying any data filters, it is usually prudent and often necessary for at least the following reasons:

1. A long established pattern in *BABAR* is to have several analyses measure the same physical quantity, but each targetting a specific subset of the data that is mostly non-intersecting with the others. If the goal of an analysis is, as is our case, to obtain an independent measurement, then it is crucial to design a filter that removes subsets that were used in the other analyses.[2]

2. Most of the data records are noise, many of which can be easily identified and discarded. This sometimes reduces the size of the dataset significantly, making it easier to work with computationally.

For the measurement of $\mathcal{R}(D^{(*)})$, table 4.1 shows the 4 approximately independent subsets of the full dataset that any single analysis can focus on. To choose amongst the samples, the analysts simply designs the appropriate data filter that selects it.

	Hadronic tagging	Semileptonic tagging
$\tau \to$ leptons	*BABAR* 2012 [1]	ongoing
$\tau \to$ hadrons	open	this analysis

Table 4.1: The 4 approximately independent subsets of the full datasets available for measuring $\mathcal{R}(D)$.

Reconstruction

One of the central components of the data filter is a procedure known as *reconstruction*, which we will define precisely in chapter 5. The key idea is that one can reconstruct almost any composite particle out of the particle list primitives provided by the *BABAR* framework; however, the quality of such a reconstruction can vary depending on whether the actual collision outcome $\omega \in \Omega$ bears any semblance to the reconstruction template. Once reconstruction has been performed, the analyst can choose a set of quality thresholds that determine whether a particular record should be kept.

The 4 samples shown in table 4.1 are selected by choosing the reconstruction template of two B mesons. The first B meson reconstruction, often termed the "tagging B", can be of two types as indicated by the columns of table 4.1. Semileptonic (hadronic) tagging targets those outcomes that have at least one B meson decaying semileptonically (hadronically). The second B meson reconstruction, often termed the "signal B", can also be of two types as indicated by the rows of table 4.1. These target those outcomes that have a $B \to D\tau\nu$ decay, and subsequently, $\tau \to$ hadrons (leptons).

Filter definition

Now that we have a set of thresholds that decides on the quality of a record, we can define the filter as a function $\mathcal{F} : \Omega \to \{\text{"yes"}, \text{"no"}\}$, where $\mathcal{F}(\omega) =$ "yes" if ω passes the thresholds and "no" otherwise.

The set $U \subseteq \Omega$ is then defined to be $\{\omega \in \Omega | \mathcal{F}(\omega) = \text{"yes"}\}$.

4.5.2 Feature extraction

In order to make any inference from the data, we must first have some set of observations recorded for each data record.

The observations that are directly available from the experiment are the electrical signals that are received due to the interactions between the particle and each pixel of the detector. While one can attempt to directly extract information out of these "images" using a few recently popular techniques, we take the more

[2] This is actually widespread across all particle physics experiments.

traditional approach of engineering features out of the particle list primitives and the reconstructed particles. As we will describe in chapter 5 we will extract a total of $d = 50$ features for each record.

The features encapsulate all the information that we could possibly extract from the experimental runs, and it is now up to us to use it as efficiently as possible in the inference procedure. For the remainder of this document, we will model the features as the d dimensional random vector $X : \Omega \to \mathbb{R}^d$.

4.6 Estimating $p_{D^{(*)}\tau_h}$

Recall that p_j for $j \in C$ is defined to be the probability that an event was drawn from event type j, given that it passed the selection criteria (see equation 4.26). Since this section is concerned with estimating p_j's from the data, we will work in the probability space $(\tilde{\Omega}, \tilde{\mathcal{U}}, \tilde{\mathbb{P}})$[3]; that is, the sample space is defined to be $\tilde{\Omega} \coloneqq U$, and the probability measure is defined to be the conditional probability $\tilde{\mathbb{P}} \coloneqq \mathbb{P}[.|U]$.

Our data set consist of N independent realizations from some continuous random variable Z, which we represent by writing $\mathcal{D} = \{z_i\}_{i=1}^N$. Since the event types partition the sample space, it must be, by the law of total probability, that the density function of Z is

$$f(z) = \sum_{j \in C} \mathbb{P}[U_j] \, f(z|U_j) \tag{4.29}$$

$$= \sum_{j \in C} p_j f_j(z) \tag{4.30}$$

where $f_j(z)$ denotes the conditional density function of Z, given that the event type is j. For reasons explained in section 4.6.2, Z is not the extracted features described in section 4.5.2, but rather a transformation of the features.

Our goal is to estimate p_j for $j = D\tau, D^*\tau$ from the dataset $\mathcal{D}$. To do so, we form the maximum likelihood estimator $\hat{p}_j$ for all $j \in C$. In particular, we solve the following optimization problem:

$$\begin{aligned} \underset{p \in \mathbb{R}^{|C|}}{\text{minimize}} \quad & -\sum_{i=1}^{N} \log \left(\sum_{j \in C} p_j f_j(z_i) \right) \\ \text{subject to} \quad & \sum_{j \in C} p_j = 1 \end{aligned} \tag{4.31}$$

Once we find the minimizer for this problem, $\hat{p}$, we can use equation 4.28 to deduce the final result. The solution of the optimation problem itself is, in principle, straightforward as it is *convex* in p, and we will solve it using CVXOPT[19]. However, before solving it, we must address the following issues:

1. We have stated that the data set came from a set of observations distributed according to some random variable X. We have not discussed what this observable is and whether there exists some transformation that will result in a more precise estimation of $\hat{p}$.
2. As it stands, the optimization problem 4.31 is posed in terms of the unknown densities $f_j(z)$'s. To make progress, we will estimate them from a different yet representative set of data and plug the result into 4.31.

The remainder of this section addresses these points.

4.6.1 Estimating event type densities

In order to fully specify the optimization problem of 4.31, we propose to estimate the unknown densities f_j for all $j \in C$ from some representative set of training data. As is typical in many other machine learning and statistical inference tasks, we begin by specifying a generic model for the density and use the training data to learn a specific instantiation.

[3] A suitable definition for $\tilde{\mathcal{U}}$ can be constructed; e.g. $\tilde{\mathcal{U}} \coloneqq \{T \cap U | T \in \mathcal{U}\}$.

Training data

Ideally, one would obtain training data that is directly sampled from f_j; unfortunately, this is often not possible in particle physics. Instead, we would choose a training set that is "sufficiently representative" of the true sample, and assess how a reasonably comprehensive set of deviations could influence the final result. The set of final result produced after applying these deviations are then publisehd as a *systematic uncertainty*[4]. The hope is that some deviation from the set of those applied will in fact be the truth; however, this is often impossible to prove. One avenue towards progress would be to at least use more principled techniques such as those used in the field *uncertainty quantification* so that we can make rigorous probabilistic statements that are based on a set of transparent assumptions.

To make progress, we use simulated data as the training set for learning the f_j's. Chapter 11 will discuss how we quantify uncertainties due to variations in the simulation model in greater detail.

Statistical model

For the task of learning the densities f_j, we use the well known kernel density estimate [20]. While the method has been known and applied for quite some time, its widespread use in large datasets[5] is limited due to the formidable computational complexity. Therefore, most implementations[6] make significant compromises in accuracy so that the computation completes in a reasonable amount of time. Of course, it is then unclear whether such rough approximations still possess the attractive theoretical properties that prompted its use in the first place.

In order to compute an accurate kernel density over our data sample[5], we implement the dual-tree algorithm proposed in [22]. To further speed up the computation, we use the GPU to accelerate the leaf computations using an algorithm similar to the one proposed in [23]. For those familiar with astrophysical N-body simulations, this is a generalization of the Barnes Hut algorithm[24] that applies to more than just central potentials of charges and masses. In the end, our algorithm not only yields a 10,000 times speedup over the naive algorithm on 2 million points, it also guarantees the accuracy to a pre-specified relative and absolute tolerance.

Our implementation also extends the algorithms cited above to perform adaptive kernel densities as well as least squares cross valdiation as presented in [20]. Without these extensions, the kernel density would not be very useful as the bandwidth choice is critical in the overall estimator performance.

4.6.2 Optimizing information efficiency in observables

In order to make a precise determination of the sample proportions p, we should make use of the information that was extracted from the data set as efficiently as possible. Specifically, we can define the random variable Z of section 4.6 to be any function of the random variable X of section 4.5.2, such that the variance of $\hat{p}$ is minimized.

Design guidelines

One might ask why a transformation is necessary at all; naively, it would seem that setting $Z = X$ would capture all the information inherent in the observations. While this might be true, it is not recommended for at least the following reasons:

1. Z should be in low dimensions. This is important so that the density function can be accurately estimated[7].

[4] In some fields, e.g. uncertainty quantification, this is termed *epistemic uncertainty*.

[5] Order of at least a million to 10's of millions.

[6] The implementation in ROOT[21] uses a gridding algorithm. While this is a common technique, it is unfortunately very approximate and generally yields large errors.

[7] The curse of dimensionality renders kernel density estimates inaccurate for high dimensions. See section 4.5 of [20] for a demonstration.

2. Z should be continuous. This is required, otherwise the density function is not well defined to begin with.

Besides stating what characteristics Z are required to have, we now derive a simple principle for designing Z. Suppose for the moment that we are interested in a modified version of the problem where there are only two event type components: signal or background. The density function can be written as follows:

$$f(z;p) = pf_S(z) + (1-p)f_B(z) \qquad z, p \in \mathbb{R} \tag{4.32}$$

Now suppose that we seek an estimator $\hat{p}$ for the parameter p. Such an estimator must satisfy the *Cramer-Rao lower bound* [25]:

$$\text{Var}\left[\hat{p}\right] \geq \frac{1}{-\mathbb{E}\left[\frac{\partial^2 \log f(z;p)}{\partial p^2}\right]} \tag{4.33}$$

where the expectation is taken over Z and the derivative is evaluated at the true value p. It is also proved in [25] that the maximum likelihood estimator in fact achieves this lower bound asymptotically.

Taking derivatives we get

$$\frac{\partial^2 \log f(z;p)}{\partial p^2} = -\left(\frac{f_S(z) - f_B(z)}{f(z)}\right)^2 \tag{4.34}$$

and therefore,

$$\text{Var}\left[\hat{p}\right] \geq \frac{1}{\mathbb{E}\left[\left(\frac{f_S(z)-f_B(z)}{f(z)}\right)^2\right]} \tag{4.35}$$

Since we are using the maximum likehood estimator in our analysis, the result in 4.35 has a very simple interpretation: to minimize the variance of $\hat{p}$, maximize the overall relative difference between the signal and background densities.

To summarize, the following are characteristics that we seek in Z:

1. Low dimensions.
2. Continuous.
3. Large expected relative difference between signal and background densities.

Statistic engineering

Of course, the estimation problem of 4.29 is not as straightforward as our toy problem in 4.32, but we will apply the principles derived from that exercise. In particular, we define $Z : \Omega \to \mathbb{R}^2$ such that

- $Z_1 = s_1(X)$, where $s_1 : X \to \mathbb{R}$ is a regression function trained to distinguish signal from background event types.
- $Z_2 = s_2(X)$, where $s_2 : X \to \mathbb{R}$ is a regression function trained to distinguish $D\tau$ from $D^*\tau$ event types.

We will defer the details of learning s_1 and s_2 to chapter 8, but for now it is important to recognize how our definition Z matches the intuition stated earlier:

1. Z has only 2 dimension.
2. Z is the output of regression functions. While this alone is far from guaranteeing continuity, this is at least a steps towards something reasonable.
3. The 2 dimensions Z focuses directly on the problem at hand: it searches for summary statistics whose primary purpose is to distinguish between signal categories from the others.

In general, finding a maximally efficient summary statistic is NP-hard. We are merely using a heuristic that is guided by some intuition as well as implementation convenience, and we do not deny the possible existence of better strategies.

4.7 Estimating $\epsilon_{D^{(*)}\tau_h}$

First express $\epsilon_{D^{(*)}\tau_h}$ in terms of quantities that we can observe or derive. For $i = D\tau_h,\, D^*\tau_h$, we have

$$\epsilon_i = \mathbb{P}\left[U_i | E_i\right] = \frac{\mathbb{P}\left[U_i \cap E_i\right]}{\mathbb{P}\left[E_i\right]} = \frac{\mathbb{P}\left[U_i\right]}{\mathbb{P}\left[E_i\right]} \tag{4.36}$$

$$= \frac{\mathbb{P}\left[U_i | E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right] + \mathbb{P}\left[U_i | E^c_{B\bar{B}}\right]\mathbb{P}\left[E^c_{B\bar{B}}\right]}{\mathbb{P}\left[E_i\right]} = \frac{\mathbb{P}\left[U_i | E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right]}{\mathbb{P}\left[E_i\right]} \tag{4.37}$$

and

$$\mathbb{P}\left[E_i | E_{B\bar{B}}\right] = \frac{\mathbb{P}\left[E_i \cap E_{B\bar{B}}\right]}{\mathbb{P}\left[E_{B\bar{B}}\right]} = \frac{\mathbb{P}\left[E_i\right]}{\mathbb{P}\left[E_{B\bar{B}}\right]} \tag{4.38}$$

$$\Rightarrow \mathbb{P}\left[E_i\right] = \mathbb{P}\left[E_i | E_{B\bar{B}}\right]\mathbb{P}\left[E_{B\bar{B}}\right] \tag{4.39}$$

The last equality of equations 4.36, 4.37, and 4.38 are true because we are considering $i = D\tau_h,\, D^*\tau_h$. Finally, we get

$$\epsilon_i = \frac{\mathbb{P}\left[U_i | E_{B\bar{B}}\right]}{\mathbb{P}\left[E_i | E_{B\bar{B}}\right]} \qquad \text{for } i = D\tau_h,\, D^*\tau_h \tag{4.40}$$

On its own, equation 4.40 is not useful since we generally do not know the proportions of each event category in real data. To make progress, we appeal to simulation where we can easily estimate the proportions by simply counting the outcomes. The catch is that we must assess the systematic uncertainty associated with the degree that the simulation represents reality. Chapter 11 will discuss how we quantify this uncertainty in greater detail.

Chapter 5

Data collection

We describe the data samples that are used in this analysis as well as how they were collected.

5.1 Data Taxonomy

There are two kinds of data. The first type are those collected by the actual running of the accelerator and the detector[1], and the second type are those produced by a computer simulation[2]

The purpose of the collider data is clear. Indeed, the whole point of our analysis is to apply a specially tailored statistical inference procedure onto the collider data in order to make a statement about $\mathcal{R}(D^{(*)})$. The purpose of the simulated data, however, is merely to enable us to construct and design this specialized inference procedure.

The following subsections describe each kind of data in greater detail.

5.1.1 Collider data

The collider data was collected by the operation of PEP-II and *BABAR* between the years of 2000-2008. As far as this analysis is concerned, this data is divided into 6 runs, each of which consists of an "On-Peak" sample and an "Off-Peak" sample. The "On-Peak" sample is the majority, and it is obtained by operating PEP-II such that collisions happen at a center of mass energy that is equal to the $\Upsilon(4S)$ mass. The "Off-Peak" sample is taken by operating PEP-II such that the center of mass energy is slightly away from the $\Upsilon(4S)$ mass, so that the majority of collision events are continuum.

As the collider operates and deposits particles into the detector, the *BABAR* software system post processes the raw signals and saves them into storage. Over the years, the software system matured and experienced several stages of significant improvements. This lead to a proliferation of various versions of the postprocessed data, which lead to a mildly complex nomenclature.

Table 5.1 and 5.2 shows the names of the post processed data samples that are used to perform this analysis.

5.1.2 Simulated data

The simulated data is divided into two types. The first is what is referred to as the "generic" MC, and the second type is referred to as the "signal" MC. For our purposes, the "generic" MC is generated such that the weighted combination of all its events will reflect our best reproduction of what an On-Peak collider run would have produced. The "signal" MC is generated to represent the outcomes of the signal event type.

[1] We will use the terms "real data" or "collider data" ta to refer to this.
[2] We will use the terms "MC" or "simulated data" to refer to this.

Collider Dataset Name	Luminosity (pb^{-1})	$N_{B\overline{B}}$
AllEventsSkim-Run1-OnPeak-R24a1	20372 ± 91	22556256.9 ± 137695.3
AllEventsSkim-Run2-OnPeak-R24a1	61321.436 ± 257.528	68438426.0 ± 413011.4
AllEventsSkim-Run3-OnPeak-R24a1	32290.437 ± 132.361	35763257.9 ± 216993.3
AllEventsSkim-Run4-OnPeak-R24a1	99606.060 ± 418.245	111429669.4 ± 671074.9
AllEventsSkim-Run5-OnPeak-R24a1	132371.946 ± 582.313	147620363.4 ± 888260.1
AllEventsSkim-Run6-OnPeak-R24a1	78327.488 ± 352.439	85194672.2 ± 513624.4
Total	424289.367 ± 853.606	$471002645.8 \pm 1319007.981$

Table 5.1: Collider On-Peak datasets. $N_{B\overline{B}}$ is the estimated number of $B\overline{B}$ pairs contained in the specified dataset; it includes both neutral and charged B pairs.

Collider Dataset Name	Luminosity (pb^{-1})	Multiplier
AllEventsSkim-Run1-OffPeak-R24a1	2563.970 ± 12.051	7.945
AllEventsSkim-Run2-OffPeak-R24a1	6869.144 ± 30.223	8.927
AllEventsSkim-Run3-OffPeak-R24a1	2443.569 ± 10.507	13.214
AllEventsSkim-Run4-OffPeak-R24a1	10015.990 ± 43.067	9.945
AllEventsSkim-Run5-OffPeak-R24a1	14276.771 ± 67.101	9.272
AllEventsSkim-Run6-OffPeak-R24a1	7752.558 ± 36.437	10.103
Total	43922.002 ± 94.096	9.660

Table 5.2: Collider Off-Peak datasets. Multiplier is the factor by which the corresponding On-Peak dataset exceeds the given Off-Peak dataset.

Table 5.3 shows the set of modes that are generated for the generic MC, and table 5.4 show the actual dataset name along with the numbers that were generated. The last column in table 5.4 shows the weight that needs to be applied to each record in order to reflect the correct counts the real On-Peak data would have had. These are computed by taking the ratio of the data luminosity to the equivalent luminosity of the simulated sample.

The signal MC is divided into two types. Both types of samples h has one B meson decaying semileptonically. The first type has the other B decaying to $D\tau$, and then $\tau \rightarrow$ hadron, while the other type decays as a $D^*\tau$, and then $\tau \rightarrow$ hadrons. Table 5.5 show the actual samples that were used. Additional details about the signal MC can be found in the appendix.

SP Mode	Mode type	Cross section (pb)
1235	B^+B^-	525.0
1237	$B^0\overline{B}^0$	525.0
1005	$c\overline{c}$	1300.0
998	uds	2090.0

Table 5.3: Cross section used to convert the sizes generic simulated data to the equivalent On-Peak dataset.

Simulated Dataset Name	Mode Type	Collisions Generated	Multiplier
SP-1235-AllEventsSkim-Run1-R24a1	B^+B^-	34878000	0.306
SP-1235-AllEventsSkim-Run2-R24a1	B^+B^-	105561000	0.305
SP-1235-AllEventsSkim-Run3-R24a1	B^+B^-	56035000	0.303
SP-1235-AllEventsSkim-Run4-R24a1	B^+B^-	166784000	0.314
SP-1235-AllEventsSkim-Run5-R24a1	B^+B^-	215168000	0.323
SP-1235-AllEventsSkim-Run6-R24a1	B^+B^-	130336000	0.316
SP-1237-AllEventsSkim-Run1-R24a1	$B^0\overline{B}^0$	34941000	0.306
SP-1237-AllEventsSkim-Run2-R24a1	$B^0\overline{B}^0$	104188000	0.308
SP-1237-AllEventsSkim-Run3-R24a1	$B^0\overline{B}^0$	57888000	0.292
SP-1237-AllEventsSkim-Run4-R24a1	$B^0\overline{B}^0$	169801000	0.307
SP-1237-AllEventsSkim-Run5-R24a1	$B^0\overline{B}^0$	215953000	0.321
SP-1237-AllEventsSkim-Run6-R24a1	$B^0\overline{B}^0$	135224000	0.304
SP-1005-AllEventsSkim-Run1-R24a1	$c\overline{c}$	55254000	0.479
SP-1005-AllEventsSkim-Run2-R24a1	$c\overline{c}$	164722000	0.483
SP-1005-AllEventsSkim-Run3-R24a1	$c\overline{c}$	88321000	0.475
SP-1005-AllEventsSkim-Run4-R24a1	$c\overline{c}$	267308000	0.484
SP-1005-AllEventsSkim-Run5-R24a1	$c\overline{c}$	344275000	0.499
SP-1005-AllEventsSkim-Run6-R24a1	$c\overline{c}$	208664000	0.488
SP-998-AllEventsSkim-Run1-R24a1	uds	176404000	0.241
SP-998-AllEventsSkim-Run2-R24a1	uds	525504000	0.243
SP-998-AllEventsSkim-Run3-R24a1	uds	276381000	0.244
SP-998-AllEventsSkim-Run4-R24a1	uds	845899000	0.246
SP-998-AllEventsSkim-Run5-R24a1	uds	1110944000	0.249
SP-998-AllEventsSkim-Run6-R24a1	uds	655152000	0.250

Table 5.4: Generic simulated data. Multiplier is the factor by which the corresponding On-Peak dataset exceeds the given simulated dataset.

Simulated Dataset Name	Mode Type	Collisions Generated	Multiplier
SP-11444-Run1-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	694000	
SP-11444-Run2-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	2044000	
SP-11444-Run3-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	1152000	
SP-11444-Run4-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	3347000	
SP-11444-Run5-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	4546000	
SP-11444-Run6-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D\tau(had)\nu$	2732000	
SP-11445-Run1-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	644000	
SP-11445-Run2-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	1937000	
SP-11445-Run3-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	955000	
SP-11445-Run4-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	3207000	
SP-11445-Run5-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	4627000	
SP-11445-Run6-R24	$B^0 \to D^{(*)}\ell\nu$, $\overline{B}^0 \to D^*\tau(had)\nu$	2349000	
SP-11446-Run1-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	651000	
SP-11446-Run2-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	1919000	
SP-11446-Run3-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	1025000	
SP-11446-Run4-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	3402000	
SP-11446-Run5-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	4276000	
SP-11446-Run6-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D\tau(had)\nu$	2685000	
SP-11447-Run1-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	750000	
SP-11447-Run2-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	1702000	
SP-11447-Run3-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	901000	
SP-11447-Run4-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	3120000	
SP-11447-Run5-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	4637000	
SP-11447-Run6-R24	$B^+ \to D^{(*)}\ell\nu$, $B^- \to D^*\tau(had)\nu$	2505000	

Table 5.5: Simulated signal data.

5.2 Event pre-screening

The first stage of the analysis is to perform a crude, yet effective screening of the records. This is mainly to reduce the computational load for the stages further downstream in the analysis, so the criteria applied here are only to filter away records that we are essentially certain to be noise.

The pre-screening criteria are as follows:

- Size of `ChargedTracks` is at most 14.
- Size of `GoodPhotonsLoose` is at most 10.
- Total charge of the event must be between $[-2, 2]$.
- Apply tag filter `BGFMultiHadron`.
- Apply tag filter `TagL3`.

5.3 Event Reconstruction

For each record in the dataset, be it real data or simulated, we perform what is referred to as "event reconstruction". This is the stage where we use the particle list primitives provided by the post processed data to see if it is possible to assemble the decay process of an outcome belonging to the signal. Records for which such a reconstruction is not possible are removed from the remainder of the analysis.

Reconstruction begins with lists of primitive particles. These are lists such as `ChargedTracks` and `ElectronKMLoose` that represent final state particles. To reconstruct a composite particle, we simply specify a particular mode from which that particle could decay along with lists of already reconstructed particles that we would like to use to substitute as its daughter particles. The reconstruction will then create and save all possible combinations of the daughters consistent with the mother hypothesis into a new list. This hierarchical reconstruction process is repeated until the $\Upsilon(4S)$ list is constructed.

A very useful way to understand event reconstruction is to model it as a directed acyclic graph. The vertices of the graph are the particles that are reconstructed, and an edge (u, v) exists if v is a daughter particle of u.

Please note that a reconstructed particle is, after all, artificial. While a successful reconstruction of a specific particle suggests that it is plausible for a true particle of said type to have actually occured during the collision, it is actually neither necessary nor sufficient. In fact, reconstructed particles are often referred to as reconstructed *candidates*, emphasizing that they are artificial.

The package that we use to perform reconstruction is `SimpleComposition`. We list below the reconstructed composite particles and the modes that they were reconstructed in reverse topological order.

1. Final state particle lists.
   - Electrons: `eCombinedSuperLoose`.
   - Muons: `muCombinedVeryLooseFakeRate`.
   - Pions: `ChargedTracks` and `GoodTracksVeryLoose`. Unless explicitly stated, the default is to use the latter.
   - Kaons: `KCombinedSuperLoose`.
   - Photons: `GoodPhotonLoose`, but with $0.01 \leq Lat \leq 0.6$.
2. Light meson lists.
   - π^0: `pi0AllDefault` and `pi0SoftDefaultMass`, after refining photon daughters using the photon list above. Unless otherwise stated, the default is to use the latter.

- $\rho^+ \to \pi^+\pi^0$: The π^+ is refined such that it cannot be a member `eKMTight` or `muBDTVeryTight`. It is also required to have mass between $[0.45, 1.09]$ GeV and a χ^2 of at least 0.001.
- K_S: `KsTight`.

3. D meson lists. All reconstructed D mesons without π^0 daughters are required to have masses within 0.06 GeV of the PDG value. For those with π^0 daughters, the masses must be withint 0.1 GeV of the PDG value.

   - $D^+ \to K^-\pi^+\pi^+$.
   - $D^+ \to K - \pi^+\pi^+\pi^0$.
   - $D^+ \to K_S K^+$.
   - $D^+ \to K_S\pi^+$.
   - $D^+ \to K_S\pi^+\pi^0$.
   - $D^+ \to K_S\pi^+\pi^-\pi^+$.
   - $D^+ \to K^-K^+\pi^+$.
   - $D^0 \to K^-\pi^+$.
   - $D^0 \to K^-\pi^+\pi^0$.
   - $D^0 \to K^-\pi^+\pi^-\pi^+$.
   - $D^0 \to K^-\pi^+\pi^-\pi^+\pi^0$.
   - $D^0 \to K_S\pi^+\pi^-$.
   - $D^0 \to K_S\pi^+\pi^-\pi^0$.
   - $D^0 \to K_S\pi^0$.
   - $D^0 \to K^-K^+$.

4. D^* meson lists. The soft pions used are required to have a center of mass 3-momentum magnitude of at most 400 MeV. Charged pions are refined from `GoodTracksVeryLoose` while neutral pions are refined from `pi0SoftDefaultMass`. Soft photons are refined from `GoodPhotonLoose`, but are required to have $Lat \leq 0.8$ and center of mass energy of at least 100 MeV and 3-momentum magnitude of at most 400 MeV.

   - $D^{*0} \to D^0\pi^0$. Require Δm to be within $[0.135, 0.175]$ GeV.
   - $D^{*0} \to D^0\gamma$. Require Δm to be within $[0.13, 0.155]$ GeV.
   - $D^{*+} \to D^0\pi^+$. Require Δm to be within $[0.135, 0.165]$ GeV.
   - $D^{*+} \to D^+\pi^0$. Require Δm to be within $[0.13, 0.15]$ GeV.
   - $D^{*+} \to D^+\gamma$. Not reconstructed, but we list it since it we only decided to remove it partway through the analysis.

5. B_{tag} meson list. Masses are required to be at most 5.2791 GeV, and the χ^2 must be at least 0.001. Furthermore, $\cos\theta_{BY}$ must be between $[-5, 1.0]$.

   - $B^+ \to \overline{D}^0 e^+$.
   - $B^+ \to \overline{D}^0 \mu^+$.
   - $B^0 \to D^- e^+$.
   - $B^0 \to D^- \mu^+$.
   - $B^+ \to \overline{D}^{*0} e^+$.
   - $B^+ \to \overline{D}^{*0} \mu^+$.

- $B^0 \to D^{*-} e^+$.
- $B^0 \to D^{*-} \mu^+$.

6. B_{sig} meson list. Masses are required to be at most 5.2791 GeV, and the χ^2 must be at least 0.001.
   - $B^+ \to \overline{D}^0 \pi^+$.
   - $B^+ \to \overline{D}^0 \rho^+$.
   - $B^0 \to D^- \pi^+$.
   - $B^0 \to D^- \rho^+$.
   - $B^+ \to \overline{D}^{*0} \pi^+$.
   - $B^+ \to \overline{D}^{*0} \rho^+$.
   - $B^0 \to D^{*-} \pi^+$.
   - $B^0 \to D^{*-} \rho^+$.

7. $\Upsilon(4S) \to B_{tag} B_{sig}$. The B daughters must conserve charge, and B's are allowed "mix". Of course, the daughters can't have overalapping final states.

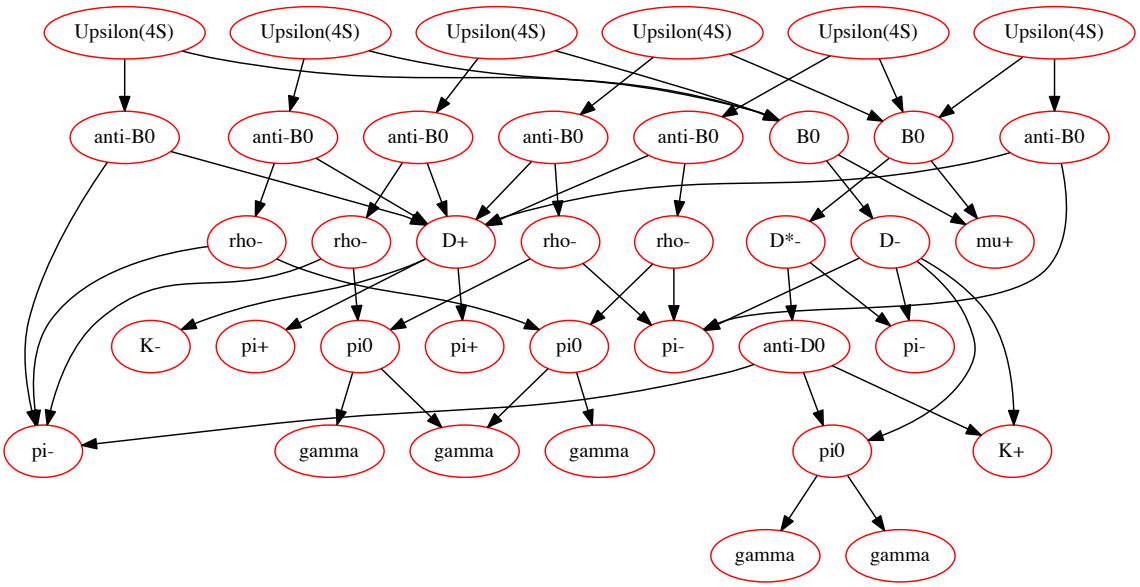


Figure 5.1: Example of event reconstruction.

Figure 5.1 shows an example of event reconstruction on an arbitrary record. We note that, in general, a reconstruction will yield many $\Upsilon(4S)$ candidates. We also emphasize again, that it is impossible to associate any specific reconstructed candidate to a true particle of the same species that could have been produced. In particular, it is impossible to declare that any specific B_{tag} candidate is "the" B meson that decayed into $B \to D\ell\nu$; such a meson need not even exist, let alone trace out the decay subtree of any particular B candidate.

The records that are discarded at this stage are precisely those whose reconstructed graph is empty.

Chapter 6

Candidate selection

Event reconstruction builds an entire graph of particle decays. The ultimate goal is to use this information to help us distinguish between signal and background.

While there are many observables that we can extract from the graph, e.g. number of tracks, number of π^0 etc, it would be convenient to elect one of the $\Upsilon(4S)$ to represent the event and extract features from its decay subtree for analysis stages further downstream.

We emphasize that we do not use this representative candidate to further reject records. Its purpose is strictly to obtain observables. One could certaintly imagine a situation where we may no longer want to be preferrential towards any candidate, but instead, apply a more democratic/global approach. Our current methodology leaves this possibility open and encourages future analyses to come up with creative ideas to capitalize on this potential.

6.1 Strategy overview

Since this single candidate will be very influential in the downstream analysis stages, we should choose a representative judiciously. Before discussing how we actually pick such a candidate, we must first discuss the criteria that renders certain candidates better than others.

Suppose a given record indeed originated from a signal collision event. Further suppose that the event reconstruction also happened to reconstruct a $\Upsilon(4S)$ candidate whose decay is "matched exactly" to the true $\Upsilon(4S)$ particle. In this case, it only makes intuitive sense that we elect this matched $\Upsilon(4S)$ candidate to represent.

Of course, the ideal scenario just described is impossible to achieve in practice, since we don't know when such a special candidate exists, let alone choose it from amongst the alternatives. Therefore, it is only practical to settle for choosing such a candidate as often as it is possible.

This naturally leads us to using supervised learning. The training data will be taken from signal MC, since we know what the generated decay process is for the simulation. We then proceed as follows:

1. For every $\Upsilon(4S)$ candidate in the training set, decide if it is matched to the underlying decay process. If it is, label it 1, and 0 otherwise.
2. Extract a set of features for the training data.
3. Learn a scoring function that indicates how likely a candidate matches.
4. Deploy the scoring function by evaluating it on every candidate in the dataset.
5. For each event, simply elect the candidate with the highest score.

6.2 Graph matching

To label a candidate in the training set, we need to define exactly what is meant when we say that a reconstructed candidate matches a true/generated particle. Before doing so, we digress briefly and introduce some graph terminology.

Definition 1 A *directed graph* is a pair $G = (V, E)$ where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$.

Definition 2 Let $G = (V, E)$ and $G' = (E', V')$ be two graphs. G is isomorphic to G' if there exists a bijection $f : V \to V'$ such that $(u, v) \in E \Leftrightarrow (f(u), f(v)) \in E'$.

The function f is only allowed to map vertices that have the same semantic information. In our application, this is the particle identity; e.g. only B^0's can be associated with B^0's, etc.

Let the reconstruction graph be $H = (V_H, E_H)$ and the true/generated decay graph be $G = (V_G, E_G)$. The subgraph of H that is isomorphic to G is the matched candidate that shall be labelled 1 for supervised learning; all others are labelled 0.

It is now clear that labelling the training data reduces to solving the famous NP-complete problem known as *subgraph isomorphism*. This is, at first glance, very discouraging news. Upon some reflection, however, we notice that there is a lot of special structure in the decay graphs that we can exploit. The structure is so constraining that we can, in fact, design algorithm 1 that solves this specific instance of subgraph isomorphism in linear time[1] by *dynamic programming*. It is easy to see that the algorithm runs in time $O(V + E)$, since each step runs either in-line with depth first search or can be computed with a single scan over the vertex or edge sets. We will, however, prove that it is correct in appendix A.

Figure 6.1 shows an example of graph matching. Figure 6.1a is the generated decay graph, but with the irrelevant particles removed; e.g. neutrinos, detector neutrons, etc. Figure 6.1b is the reconstruction graph that is built for this record. Figure 6.1c shows the output of the graph matching. The highlighted vertices are the candidates that matches to a particle in the generated decay graph.

It turns out that this algorithm is quite general and applies to most particle reconstruction matching problems commonly encountered. The standard method in BABAR that addresses the same problem is to use what is called "truth matching". This methodology is pretty well defined for final state particles, but it was not clear to me that the composite and unobservable particles were handled correctly for our application.

6.3 Feature selection

Starting from this section until the end of this chapter, we will refer to candidates that are matched as "signal" and others as "noise" unless stated otherwise. This is not to be confused with the signal and background event types referred to elsewhere in this document; we temporarily redefine this term since this is more convenient terminology when discussing supervised learning.

To proceed with supervised learning, we must extract a set of features useful for differentiating signal from noise. They are the following:

`mmiss2prime`, `eextra`, `tag_lp3`, `tag_cosby`, `tag_costhetadl`, `tag_dmass`, `tag_deltam`, `tag_costhetadsoft`, `tag_softp3magcm`, `sig_hp3`, `sig_cosby`, `sig_costhetadtau`, `sig_vtxb`, `sig_dmass`, `sig_deltam`, `sig_costhetadsoft`, `sig_softp3magcm`, `sig_hmass`, `sig_vtxh`, `tag_isbdstar`, `sig_isbdstar`, `tag_dmode`, `tag_dstarmode`, `sig_dmode`, `sig_dstarmode`.

The features we listed above are a subset of what we will be discussing in much greater detail in chapter 7, and so we will defer their descriptions and issues such as how well simulation represents real data until then. In the present context, we simply want to make the point that there is abundant information in the dataset that we can utilize to better perform signal and noise differentiation. This can be seen in figure 6.2, which shows the density functions for signal and noise for a select few features.

[1] Actually, we solve a special case that we call *subtree isomorphism*.

Algorithm 1 PARTICLE-GRAPH-MATCHING(G, H, M)

Input:

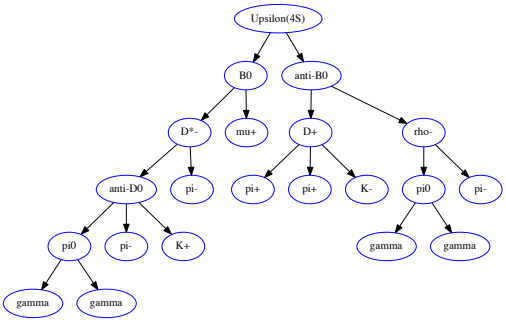
- G: Generated particle decay graph.
- H: Reconstructed particle decay graph.
- M: Array of length $|V_H|$. $M[u] = u'$ if vertex $u \in V_H$ has no out-edges and has been pre-determined to be matched to vertex $u' \in V_G$ according to the detector deposition pattern. $M[u] = null$ otherwise.

Output:

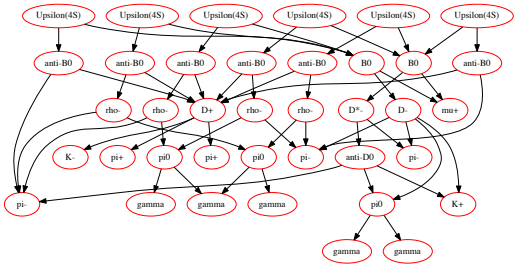
- Vertex $u' \in V_H$ if the root vertex of G is tree-isomorphic to u'. If there isn't such a vertex, return $null$.

Procedure:

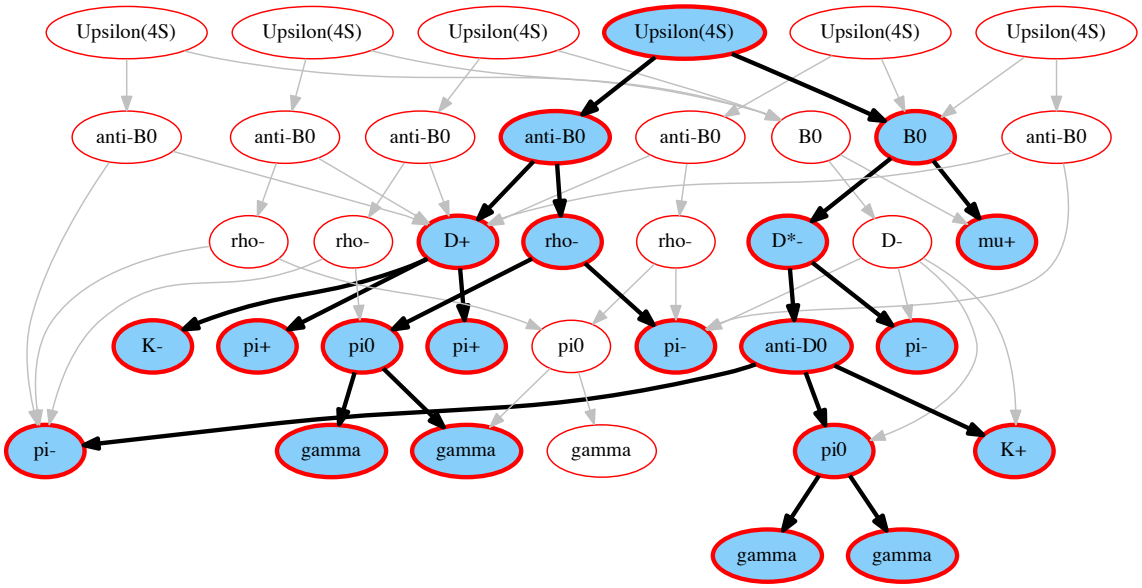
1. Loop through V_G and find the root vertex r. It is the one without any in-edges.
2. Run depth first search on H. Perform following checks when a non-terminal $u' \in V_H$ is colored black:
   (a) For each daughter v' of u', check that $M[v']$ is non-null.
   (b) For each of $M[v']$, check that they correspond to the same mother. Denote this vertex as m.
   (c) Check that m has the same out-degree as u'.
   (d) Check that m has the same particle identity as u'.
   (e) If all of the conditions above are met, set $M[u'] = m$, otherwise $null$.
3. Loop through all vertices $u \in V_H$, and find whether any is such that $M[u] = r$. If so, return u, otherwise return $null$.



(a) Generated decay graph with irrelevant particles removed.

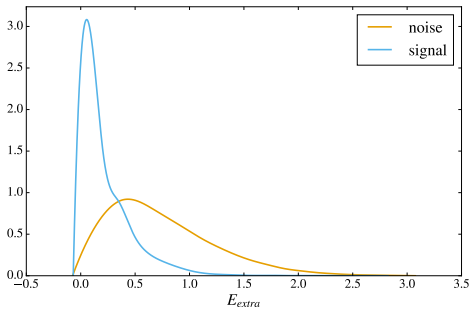


(b) The corresponding reconstruction graph over this record.

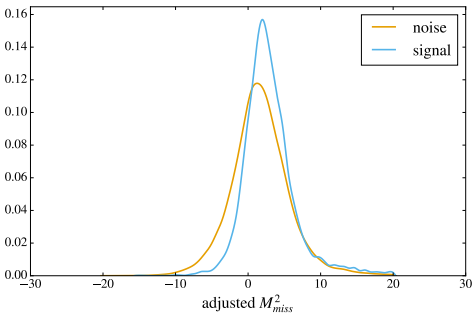


(c) The corresponding reconstruction graph over this record.

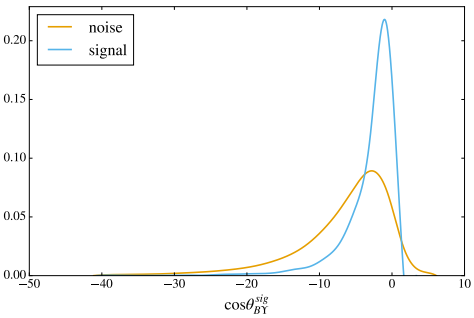
Figure 6.1: Graph matching example.



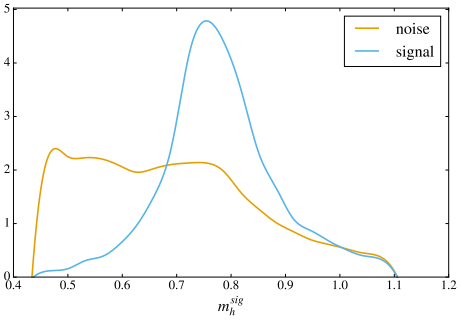
(a) E_{extra}.



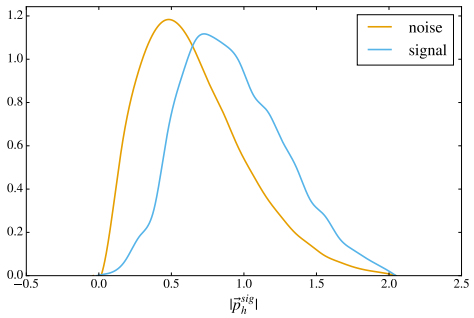
(b) Adjusted M^2_{miss}



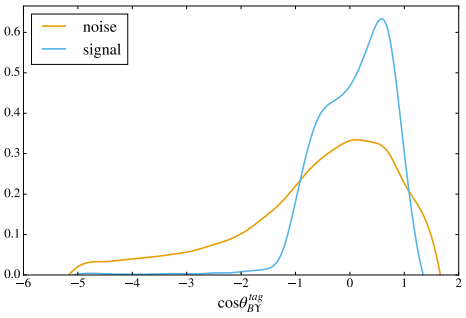
(c) $\cos\theta^{sig}_{BY}$



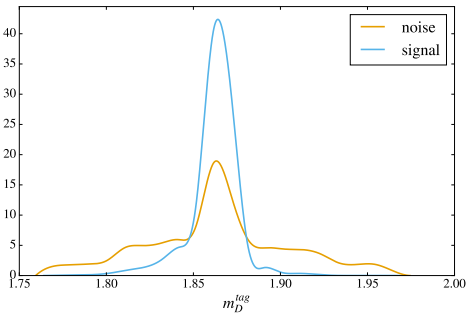
(d) m_h



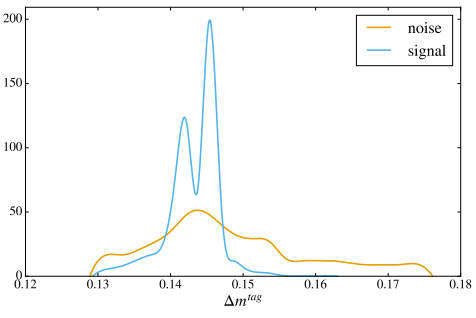
(e) $\|\vec{p}_h\|$



(f) $\cos\theta^{tag}_{BY}$



(g) m_D^{tag}



(h) Δm^{tag}

Figure 6.2: Density functions for signal and noise in select features

	minimum E_{extra}	maximum candidate score
total possible	5471	5471
number chosen	3102	4434
efficiency	0.57	0.81

Table 6.1: "Total possible" is the number of collision events that has a matched candidate that can be chosen. "Number chosen" is the number of those collision events that the specified criteria was able to correctly identify the matched candidate. "Efficiency" is simply the ratio between the rows above.

6.4 Supervised learning

Now that we have all the ingredients required for machine learning, we proceed to learn a scoring function that maps a point in feature space to a real number in $[0, 1]$ that estimates the probability that such a point is signal.

As will be shown in the results section, the function that we learn by simply applying the standard recommendations in data preporcessing as well as by applying a powerful yet easy to tune model would have already gotten large improvements compared to the established methodology of choosing the lowest E_{extra} candidate. Since the improvements are so large, we decided not to spend more time on performance tuning, and simply move on to the next stage.

The training data consists of about $600K$ records, each containing about 19 numerical features and 6 categorical features.

For data preprocessing, we mean center each numerical feature, as well as scaling their variances to 1. For each categorical feature, we simply convert it to the one-hot encoding.

For the training itself, we use the random forest [26] and take most of the default settings in scikit-learn [27]. The only setting that differ from the defaults is to take the number of trees to 1000. This might sound absurd at first glance, but we decided to do this because the way random forests determine a probability is by voting among the trees. More trees give more precision, which implies less arbitrary tie breaking when we choose a candidate. The only problem with such a large number of trees is the computation time and the space resources; however, neither of these pose a big problem for our machines.

6.5 Results

After training the scoring function, we deploy it onto the test sample, which also consists of about $600K$ records from the signal MC, where those labeled as noise are records that were not truth matched.

Figure 6.3 shows the density function for the score function for the signal and noise categories. It shows clear differences between the two categories; furthermore, it visually has better separation compared to any single feature shown in the plots of figure 6.2.

The comparison above alone is not enough to demonstrate the efficacy of our score function. To show that we are actually doing better, we select the best candidate for the test sample with two methods. The first chooses the candidate by using the lower E_{extra} candidate, while the second chooses the candidate that scores the highest with our function. The result is shown in table 6.1, which shows that our scoring function chooses 40% more matched candidates than the lowest E_{extra} criterion.

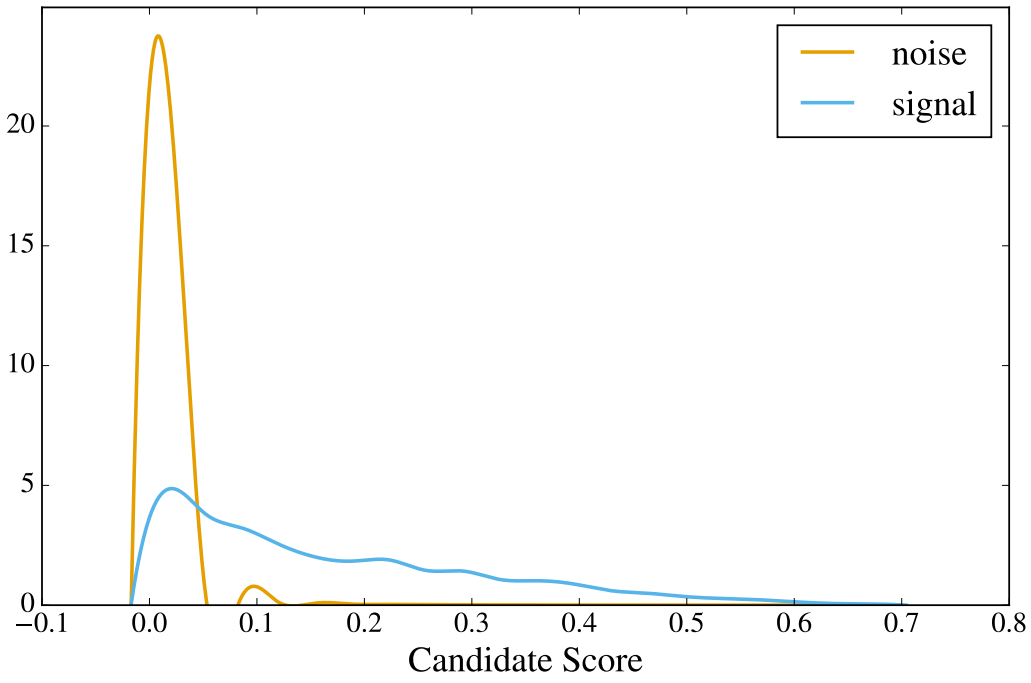


Figure 6.3: Density of the score function for signal and noise categories.

Chapter 7

Feature extraction

This chapter describes the features that we extract from each record of the data sample for the purpose of signal detection (chapter 8).

The goal of this chapter is two-fold. The first is to visualize each feature so that we can check that they are behaving correctly according to our physical intuition. The second is to check whether the features are correctly modelled by the simulation; this is important since the later stages of the analysis relies heavily on the simulation as input.

7.1 Feature description

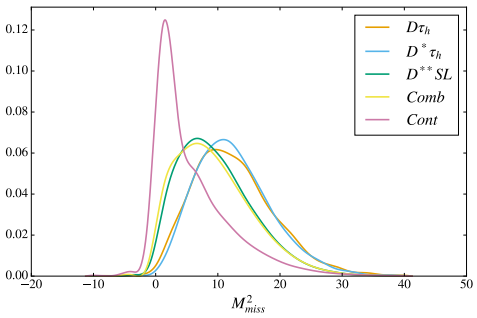
The following is a list of the features that we extract from each event record. Each item is a single feature that lists its name, alias (parenthesized), and a brief description of what it is.

- $n\Upsilon$ (`ny`): Number of $\Upsilon(4S)$ candidates.
- N_{track} (`ntracks`): Number of `GoodTracksVeryLoose`.
- R_2 All (`r2all`): Second Fox-Wolfram moment.
- Candidate Score (`cand_score`): Candidate score from chapter 6.
- M^2_{miss} (`mmiss2`): Missing mass squared.
- E_{extra} (`eextra`): Leftover photon energy.
- $\cos\theta_T$ (`costhetat`): cos of the angle between the thrust and the beam momentum (double check?).
- $|\vec{p}^{\,tag}_{\ell}|$ (`tag_lp3`): 3-momentum magnitude of the B_{tag}'s lepton.
- $\cos\theta^{tag}_{BY}$ (`tag_cosby`): cos of the angle between the 3-momentum of the B_{tag} and the 3-momentum sum of its D and lepton daughters.
- $\cos\theta^{tag}_{D\ell}$ (`tag_costhetadl`): cos of the angle between the 3-momenta of the D and the lepton daughter of the B_{tag}.
- m^{tag}_D (`tag_dmass`): Mass of the B_{tag}'s D meson daughter.
- Δm^{tag} (`tag_deltam`): Δm of the B_{tag}'s D^* meson daughter if it exists.
- $\cos\theta^{tag}_{Dsoft}$ (`tag_costhetadsoft`): cos of the angle between the D^* mesons' daughters.
- $|\vec{p}^{\,tag}_{soft}|$ (`tag_softp3magcm`): 3-momentum magnitude of the D^*'s soft daughter.

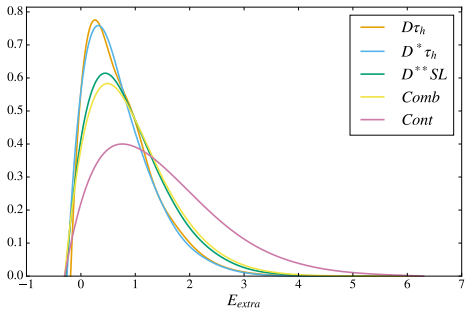
- $|\vec{p}_h^{sig}|$ (`sig_hp3`): 3-momentum magnitude of the B_{sig}'s hadron daughter.
- $\cos\theta_{BY}^{sig}$ (`sig_cosby`): cos of the angle between the 3-momentum of the B_{sig} and the 3-momentum sum of its D and lepton daughters.
- $\cos\theta_{D\tau}^{sig}$ (`sig_costhetadtau`): cos of the angle between the 3-momentum of the B_{sig} and the 3-momentum sum of its D and hadron daughters.
- χ^2_{sigB} (`sig_vtxb`): χ^2 of the B_{sig}'s vertex fit.
- m_D^{sig} (`sig_dmass`): Mass of the B_{sig}'s D meson daughter.
- Δm^{sig} (`sig_deltam`): Δm of the B_{sig}'s D^* meson daughter if it exists.
- $\cos\theta_{Dsoft}^{sig}$ (`sig_costhetadsoft`): cos of the angle between the D^* mesons' daughters.
- $|\vec{p}_{soft}^{sig}|$ (`sig_softp3magcm`): 3-momentum magnitude of the D^*'s soft daughter.
- m_h^{sig} (`sig_hmass`): Mass of the B_{sig}'s hadron daughter, if it exists.
- χ^2_{sigh} (`sig_vtxh`): χ^2 of the B_{sig}'s composite hadron daughter, if it exists.
- tag D mode (`tag_dmode`): The mode in which the B_{tag}'s D daughter is reconstructed. The modes are assigned an integer code in the same order listed in 5.3.
- tag D^* mode (`tag_dstarmode`): The mode in which the B_{tag}'s D^* daughter is reconstructed. The modes are assigned an integer code in the same order listed in 5.3.
- sig D mode (`sig_dmode`): The mode in which the B_{sig}'s D daughter is reconstructed. The modes are assigned an integer code in the same order listed in 5.3.
- sig D^* mode (`sig_dstarmode`): The mode in which the B_{sig}'s D^* daughter is reconstructed. The modes are assigned an integer code in the same order listed in 5.3.
- tag ℓ electron PID (`tag_l_epid`): B_{tag}'s lepton daughter's electron PID level.
- tag ℓ muon PID (`tag_l_mupid`): B_{tag}'s lepton daughter's muon PID level.
- sig h electron PID (`sig_h_epid`): B_{sig}'s hadron daughter's electron PID level.
- sig h muon PID, (`sig_h_mupid`): B_{sig}'s hadron daughter's muon PID level.
- Is $B_{tag} \to D^*$? (`tag_isbdstar`): Flag to indicate whether B_{tag} is reconstructed as a semileptonic D or D^* decay.
- Is $B_{sig} \to D^*$? (`sig_isbdstar`): Flag to indicate whether B_{sig} is reconstructed as a semileptonic D or D^* decay.

7.2 Event type density functions

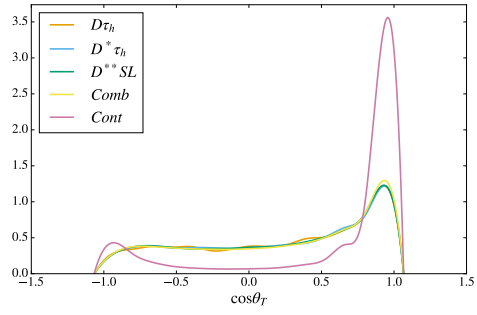
Figures 7.1, 7.2, 7.3, and 7.4 show the estimated density functions (based on a smoothing spline) of every event type for each feature listed above. We note that all of the features seem to behave according to our physical intuition, and most of them contain at least some information that can tell certain event types from others.



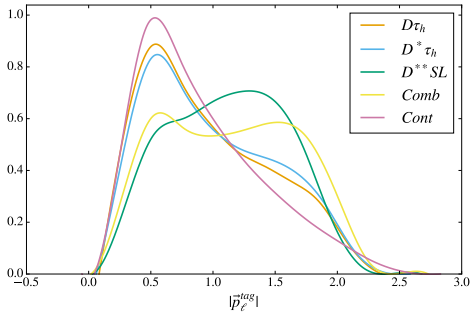
(a) M^2_{miss}.



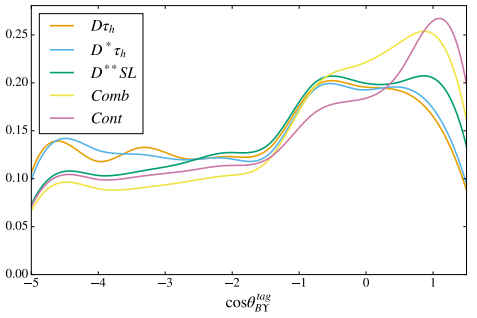
(b) E_{extra}.



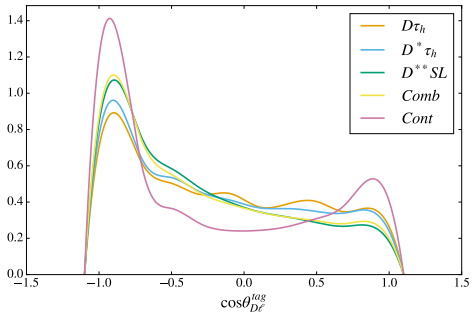
(c) $\cos\theta_T$.



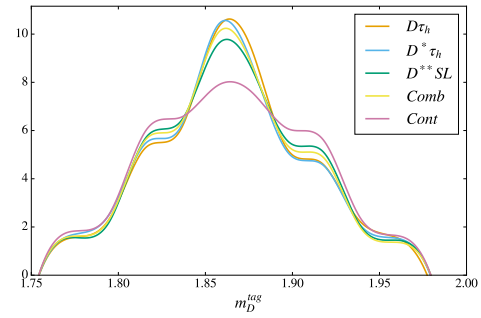
(d) $|\vec{p}^{tag}_\ell|$.



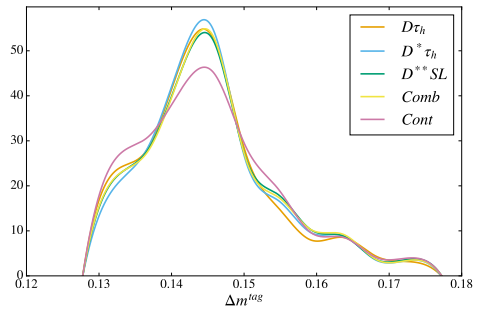
(e) $\cos\theta^{tag}_{BY}$.



(f) $\cos\theta^{tag}_{D\ell}$.

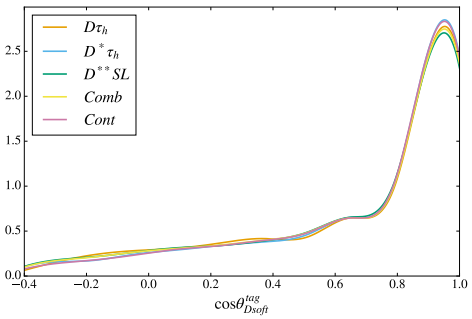


(g) m^{tag}_D

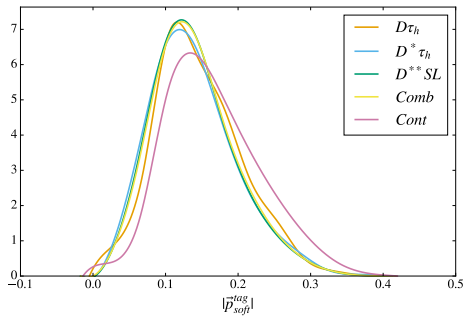


(h) Δm^{tag}.

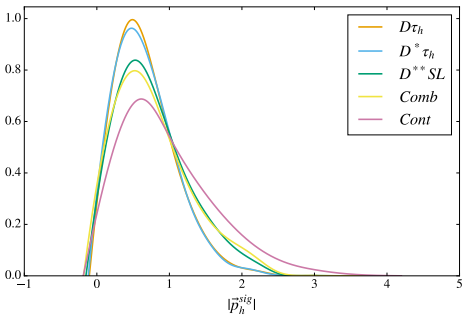
Figure 7.1: Feature density functions for each event type.



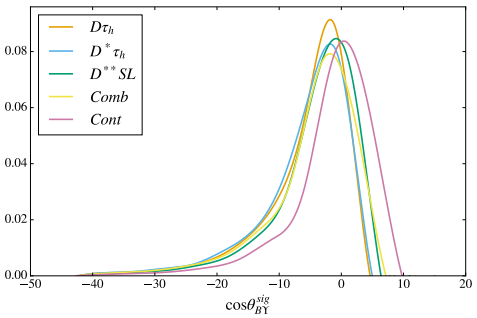
(a) $\cos\theta^{tag}_{Dsoft}$.



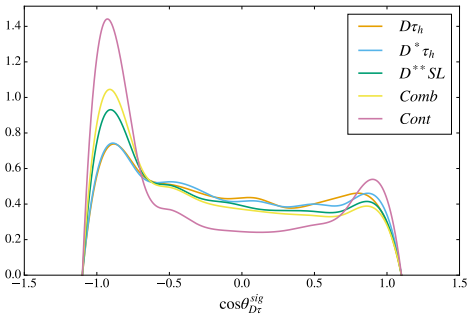
(b) $|\vec{p}^{tag}_{soft}|$.



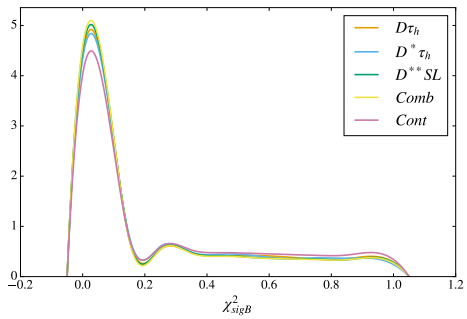
(c) $|\vec{p}^{sig}_{h}|$.



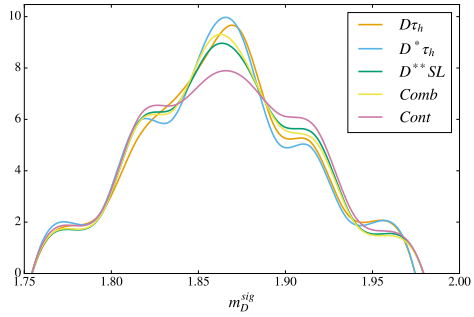
(d) $\cos\theta^{sig}_{BY}$.



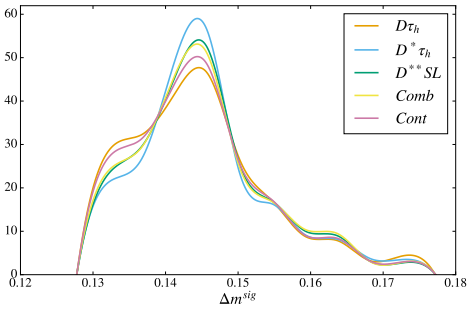
(e) $\cos\theta^{sig}_{D\tau}$.



(f) $B_{sig}\,\chi^2$.

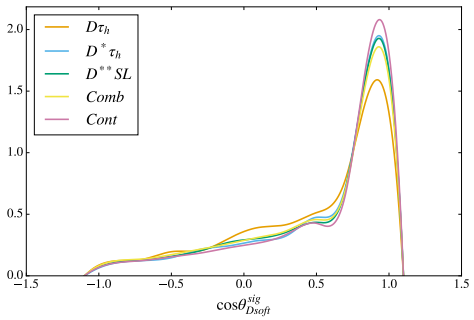


(g) m^{sig}_{D}

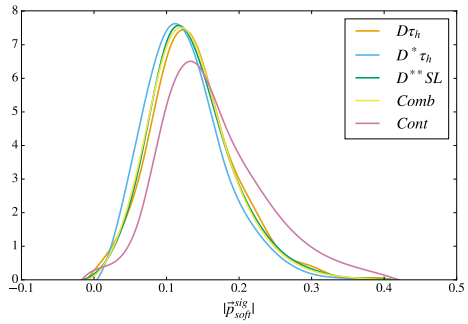


(h) Δm^{sig}.

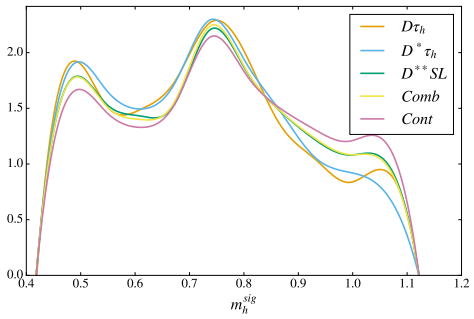
Figure 7.2: Feature density functions for each event type.



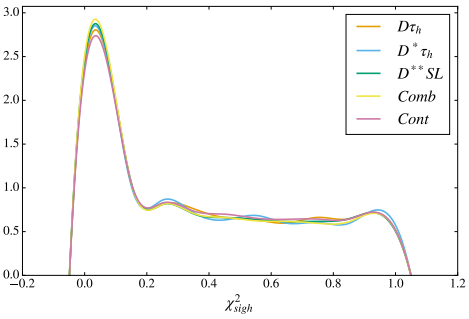
(a) $\cos\theta^{sig}_{Dsoft}$.



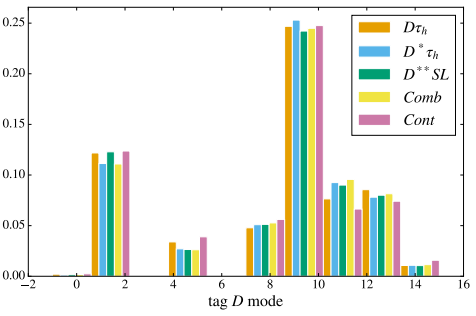
(b) $|\vec{p}^{sig}_{soft}|$.



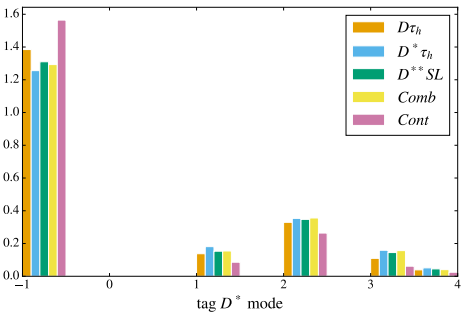
(c) m_h.



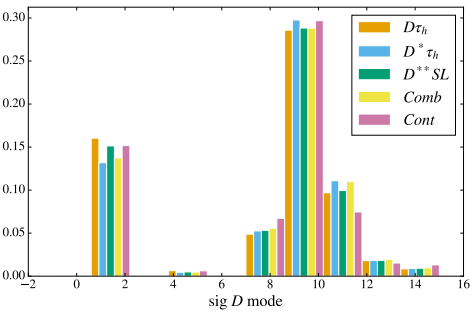
(d) $h\,\chi^2$.



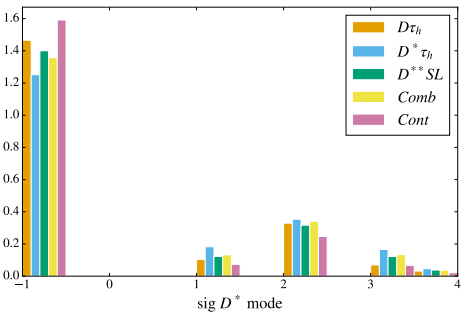
(e) B_{tag} D mode.



(f) B_{tag} D^* mode.

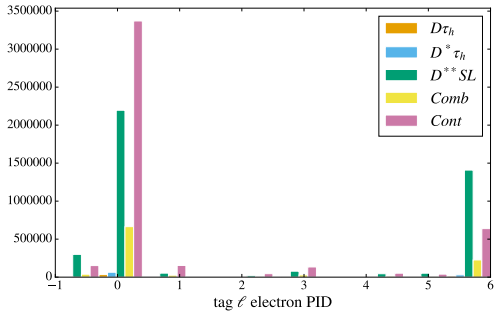


(g) B_{sig} D mode.

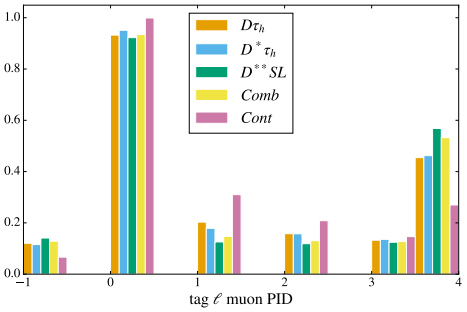


(h) B_{sig} D^* mode.

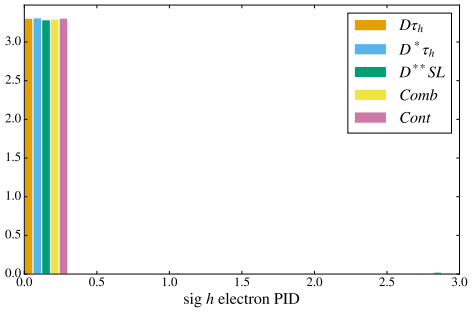
Figure 7.3: Feature density functions for each event type.



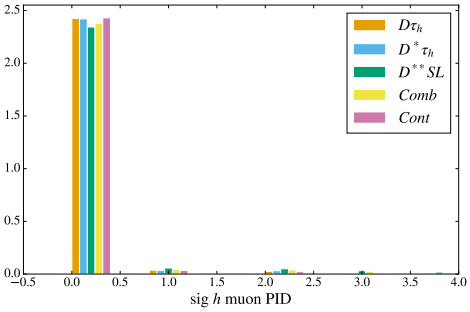
(a) B_{tag}'s ℓ daugther electron PID level.



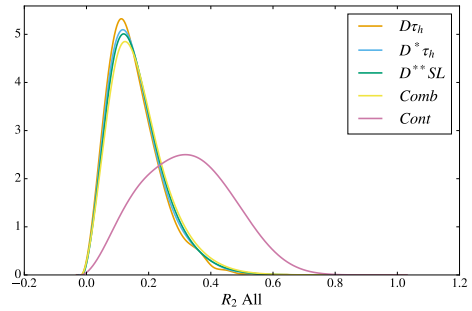
(b) B_{tag}'s ℓ daugther muon PID level.



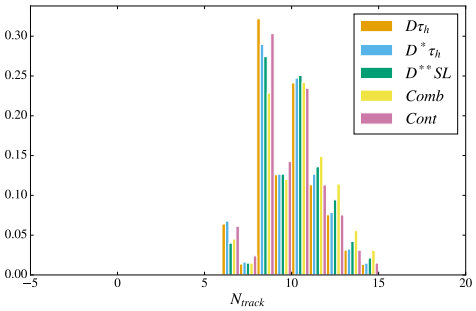
(c) B_{sig}'s hadron daugther electron PID level.



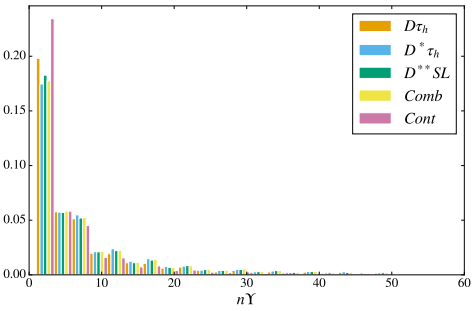
(d) B_{sig}'s hadron daugther muon PID level.



(e) $R2$.



(f) Ntracks.



(g) $n\Upsilon$.

Figure 7.4: Feature density functions for each event type.

7.3 Simulation fidelity

Figures 7.5, 7.6, 7.7, and 7.8 show the comparisons between data and MC for each feature.

The black points are based on 1% of the On-Peak data that was sampled uniformly at random; they reserved strictly for this comparison, and exclud them from the remainder of the analysis.

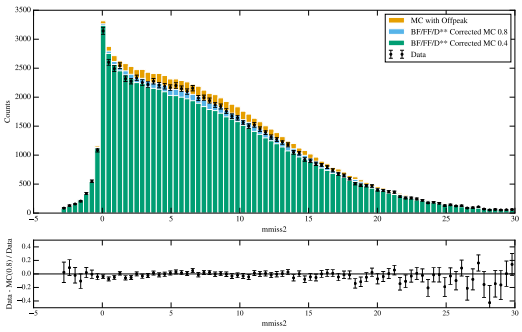
The colored histograms are based primarily on 4% of the generic MC, but scaled such that its luminosity is equivalent to the black points. We also post processed the generic MC before making the histograms with the following:

1. Remove the continuum component of generic MC and replace it with Off-Peak data that has been scaled to the correct luminosity.

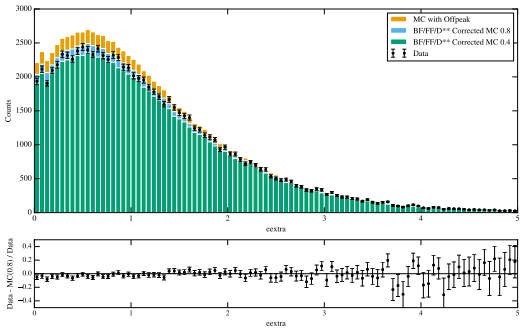
2. Reweighted the branching fractions of the most frequently occuring B decay modes to the most recent PDG values. This will be discussed in more detail in section 11.3.

In addition, we also re-weight the branching fraction of the B semileptonic $D**$ modes by a few factors to show the impact of its branching fraction uncertainty on the overall normalization. One might at first think that scaling down the D^{**} branching fraction by a factor of 0.8 may seem like a lot, but we note that even the branching fractions of B semileptonic decay modes to $D^{(*)}$ were 10% higher in `DECAY.DEC` than those in the PDG.

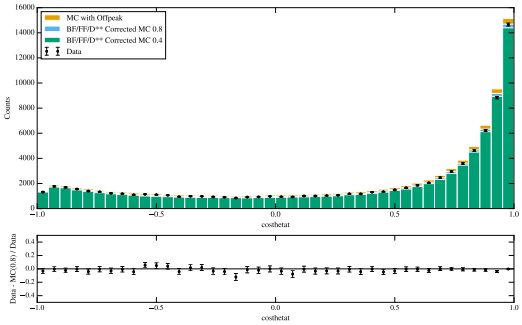
Aside from the uncertainty in the MC normalization, we note that the overall shape of each feature seem to agree reasonably well between data and MC.



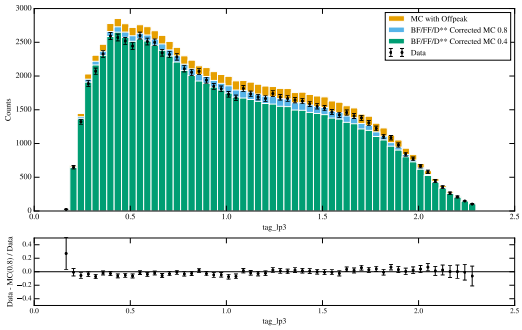
(a) M^2_{miss}.



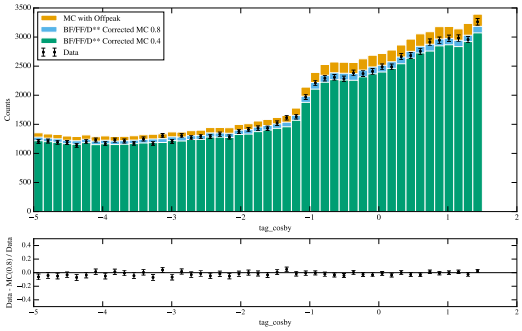
(b) E_{extra}.



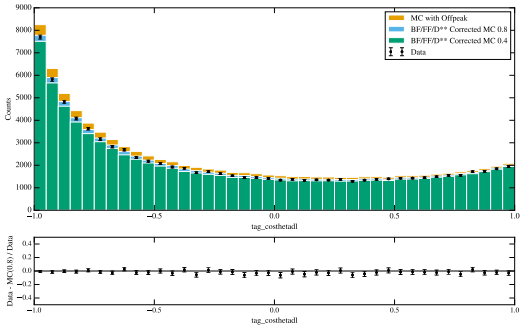
(c) $\cos\theta_T$.



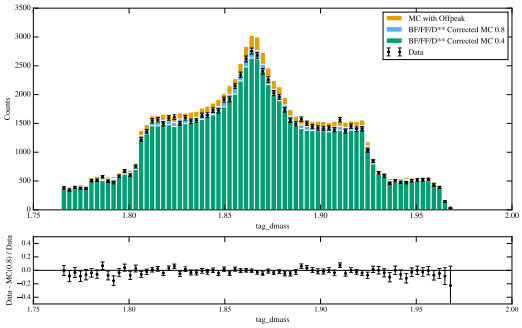
(d) $|\vec{p}_\ell^{\,tag}|$.



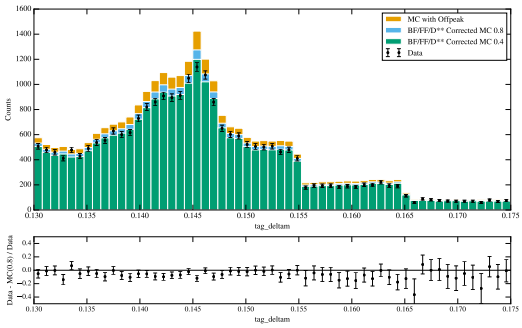
(e) $\cos\theta^{tag}_{BY}$.



(f) $\cos\theta^{tag}_{D\ell}$.

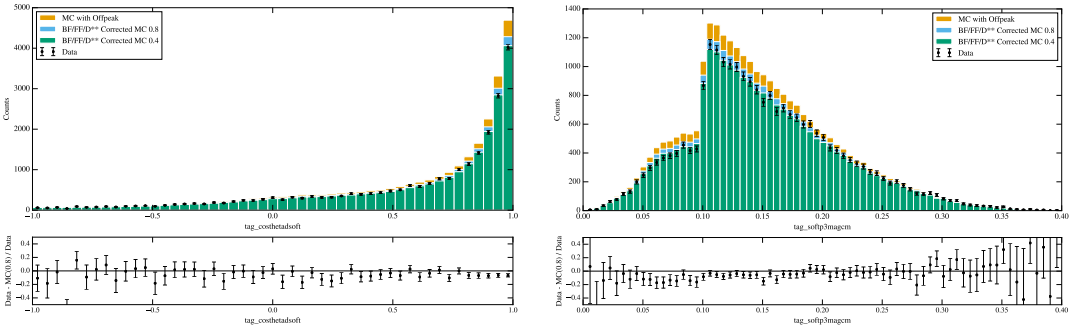


(g) m^{tag}_D



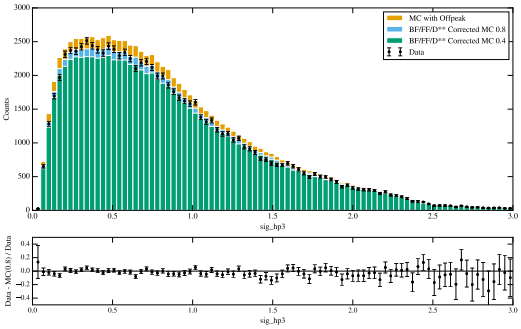
(h) Δm^{tag}.

Figure 7.5: Comparisons between data and MC for each event type.

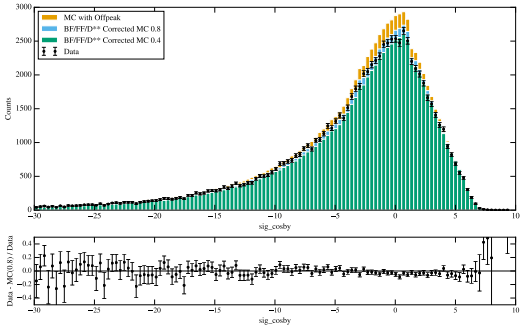


(a) $\cos\theta^{tag}_{Dsoft}$.

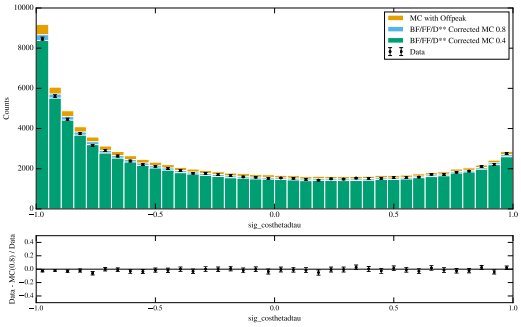
(b) $|\vec{p}^{tag}_{soft}|$.



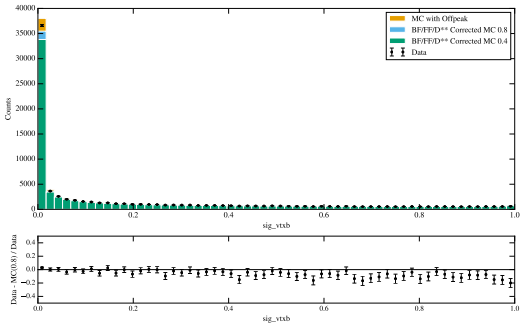
(c) $|\vec{p}^{sig}_{h}|$.



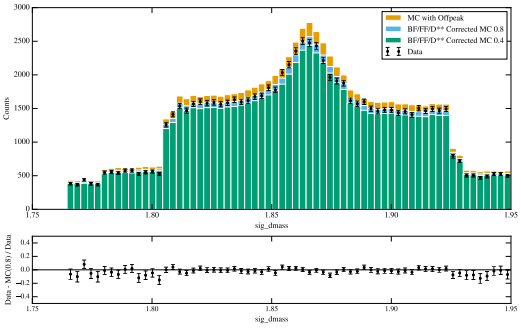
(d) $\cos\theta^{sig}_{BY}$.



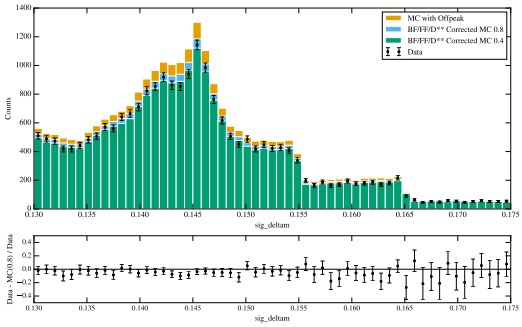
(e) $\cos\theta^{sig}_{D\tau}$.



(f) $B_{sig}\,\chi^2$.

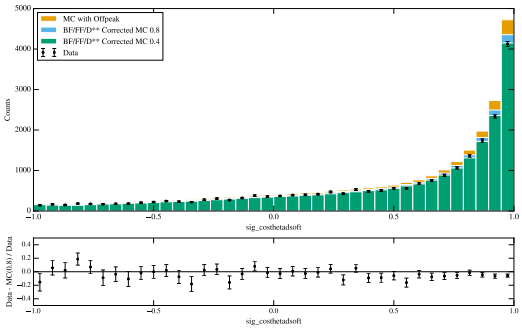


(g) m^{sig}_{D}

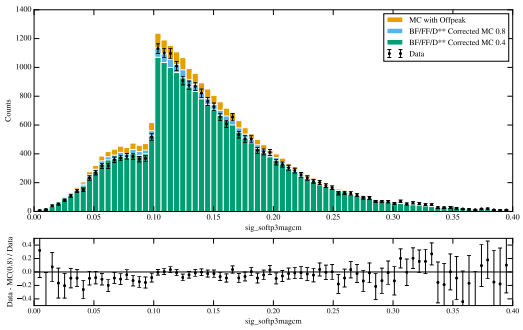


(h) Δm^{sig}.

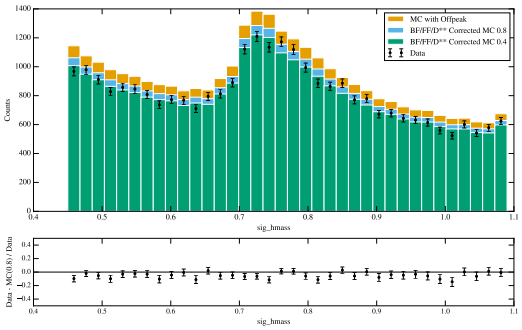
Figure 7.6: Comparisons between data and MC for each event type.



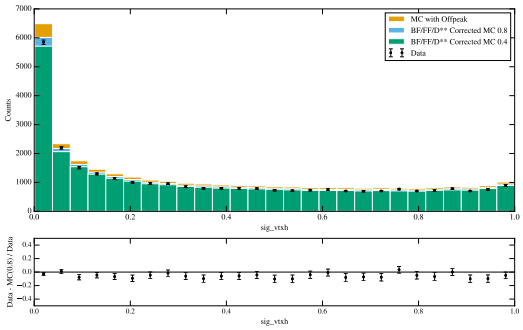
(a) $\cos\theta^{sig}_{Dsoft}$.



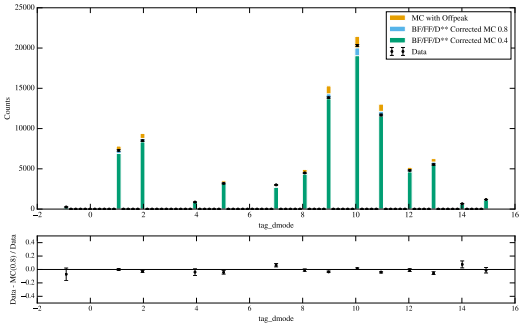
(b) $|\vec{p}^{sig}_{soft}|$.



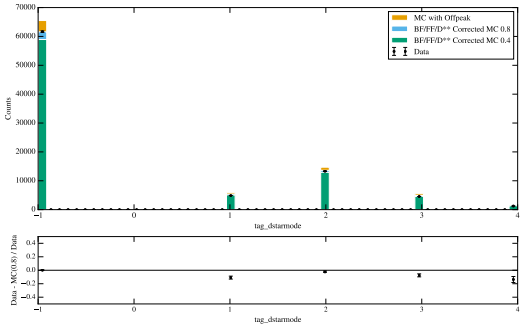
(c) m_h.



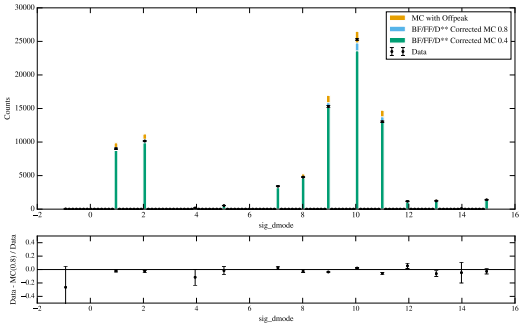
(d) $h\,\chi^2$.



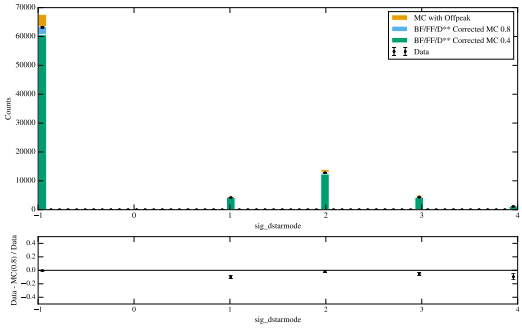
(e) B_{tag} D mode.



(f) B_{tag} D^* mode.

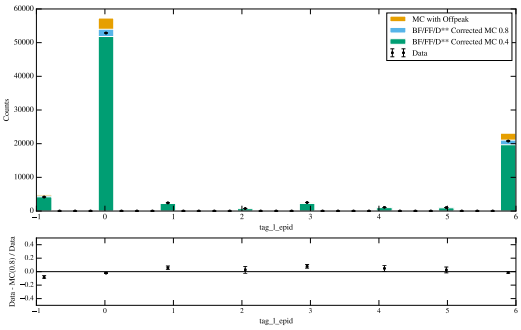


(g) B_{sig} D mode.

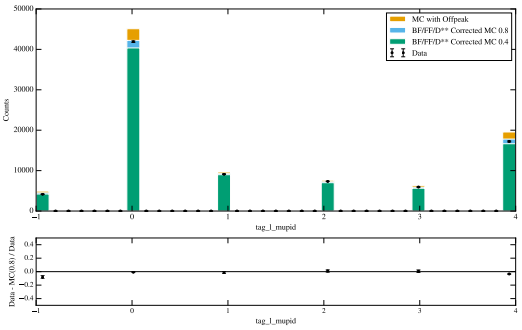


(h) B_{sig} D^* mode.

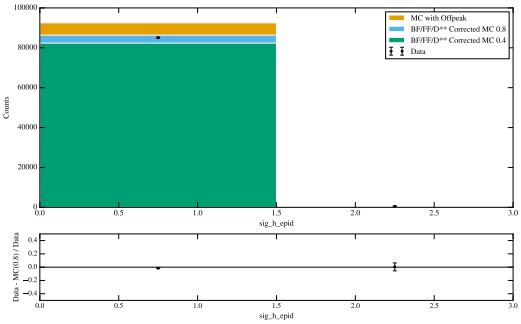
Figure 7.7: Comparisons between data and MC for each event type.



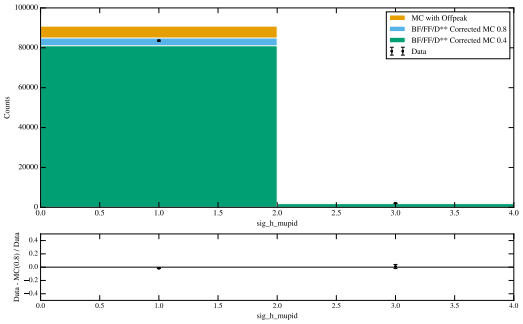
(a) B_{tag}'s ℓ daugther electron PID level.



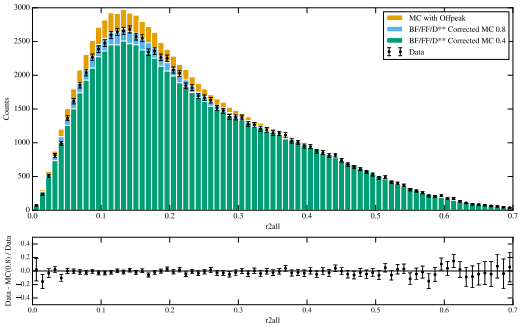
(b) B_{tag}'s ℓ daugther muon PID level.



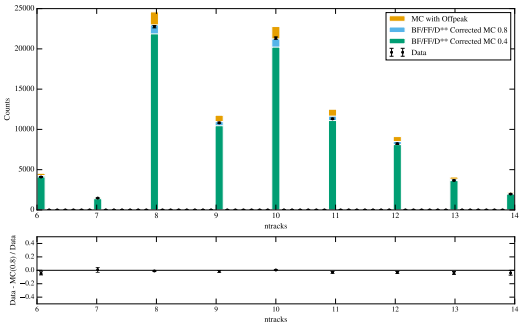
(c) B_{sig}'s hadron daugther electron PID level.



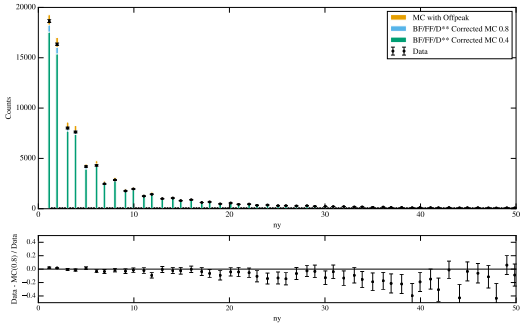
(d) B_{sig}'s hadron daugther muon PID level.



(e) $R2$.



(f) Ntracks.



(g) $n\Upsilon$.

Figure 7.8: Comparisons between data and MC for each event type.

Chapter 8

Signal Detection

The discussion in section 4.6.2 justified the importance of finding a low dimensional representation of the dataset in order to minimize the statistical uncertainty of the measurement. We have also decided to use 2 statistics to summarize the dataset: the first statistic (Z_1) differentiates between signal from background event types, and the second (Z_2) statistic differentiates between $D\tau$ and $D^*\tau$ signal.

This chapter describes how we actually obtain these two statistics using supervised learning. We also assess their performances and demonstrate that what we have is very close to optimal given the resources that are readily available.

We construct each of Z_1 and Z_2 in two steps:

1. Use supervised learning to train a set of models that solves the appropriate classification problem.
2. From amongst these models, choose one that is most suitable, and perform any final transformations.

The first two sections describe how we train various flavors of classifiers for designing Z_1 and Z_2 respectively. The last section discusses our specific choice and how we transform those scores into Z_1 and Z_2.

8.1 Signal detector

This learning problem is relevant to defining the statistic Z_1, which concerns distinguishing between the signal and background event types.

The training and testing data used both contain approximately $600K$ records, uniformly sampled from the generic MC, and consists of the ~ 34 features listed in chapter 7. The training labels are such that the signal categories are labelled 1, and 0 otherwise.

8.1.1 Data preprocessing and learning models

The data was first preprocessed with the following steps:

1. Missing values are mean imputed, but their absence is indicated with a separate boolean feature.
2. Each feature is mean centered and scaled to unit variance.
3. Categorical features are one-hot encoded.

The preprocessed data is then used as input for training the learning models listed below in scikit-learn [27]:

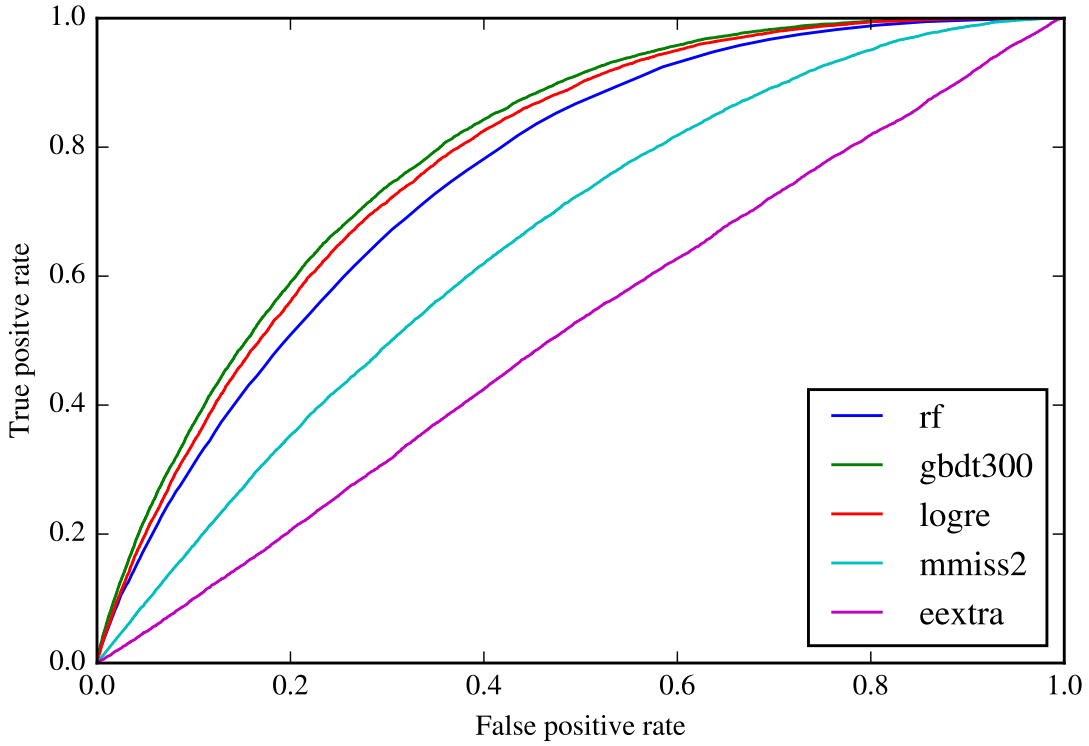


Figure 8.1: ROC curve for the Z_1 learners.

- Random forests: With the exception of taking the number of random trees to be 1000, we take the default setting in scikit-learn. The reason for choosing such a large number of trees is the same as those in candidate selection problem; that is, we want larger precision in the probability outputs.
- (Gradient) boosting: With the exception of taking the number of boosting iterations to 300, we take the default settings. In this case, we had to take care that we were not overfitting.
- Logistic regression: Take all the defaults in scikit-learn.

As you can see, these models are all relatively easy to tune; in fact, they were chosen precisely for this reason. The first two models are known to perform well on a wide range of problems while the last one is less expressive but far easier to understand.

8.1.2 Analysis

Recall that equation 4.35 would have been our preferred metric from which to measure the performance of each learner. However, it is not readily available for the learning algorithms to optimize directly; we thus use the area under the ROC curve as an alternative since it measures similar effects on an intuitive level. Of course, using such a proxy only estimates the actual performance. We are thus using it only as a device to gain some insight into the problem and to help us make the final choice.

Figure 8.1 shows the ROC curves for each of the trained models. All curves corresponding to learned models are reasonably close to each other with the boosted model having a slight edge. While there is good evidence against the random forest model solely based on the output of this figure, we have actually more

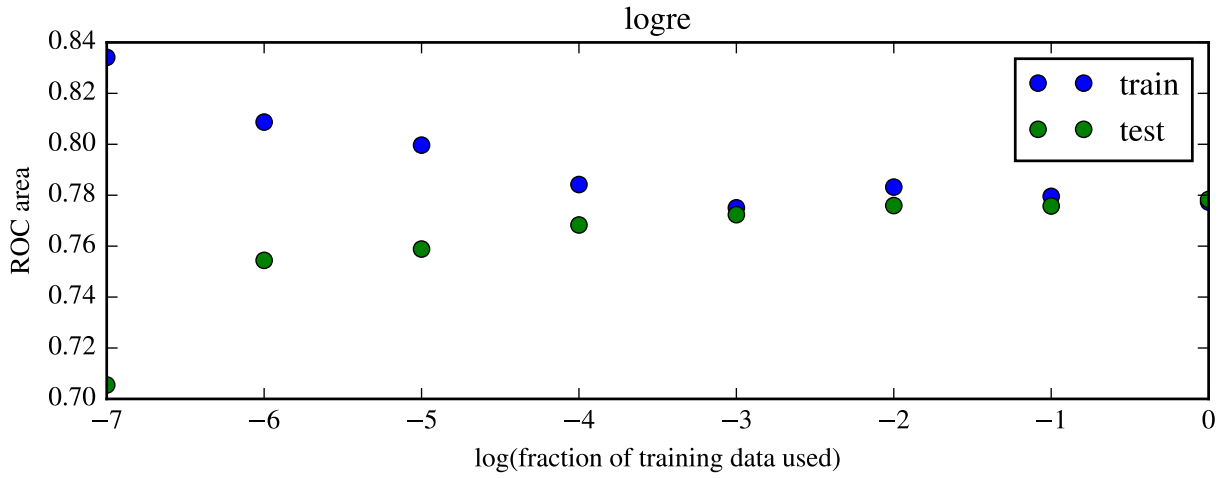


Figure 8.2: Learning curve for the logistic regression learner.

compelling reason to do so. The random forest's probability output precision is only as high as the number of trees allows; in particular, we get only 3 digits of precision with 1000 trees. This might not seem like a very big problem from the standpoint of classification accuracy, but it matters a lot more when we try to learn its density. The problem is that the discretization effect can prevent the kernel density cross validation from finding the ideal bandwidth, thus compromising the quality of the maximum likelihood and thereby signal extraction.

Figure 8.1 also includes the ROC curves for using some of the most powerful direct physics observables as discriminators. One can see that, by simply using a learning model, we are doing at least 2x better than the case of just using M^2_{miss}. It is also interesting to note that using just E_{extra} turns out to be only slightly better than random guessing.

Figure 8.1 shows that all of the learners have similar performances, which seems surprising at first since the logistic regression is a far simpler model compared to the other two models; in fact, it even does better than the random forest for this specific problem. This could happen for several reasons, but we suspect that it is due to the abundance of the training data that makes even "simple" models perform well.

This is quantified by what is known as the *VC bound* [28][29] and we can actually check whether we are in this regime by plotting the so-called *learning curve*. These curves are plotted for the logistic regression learner and is shown in figure 8.2. It shows the training and testing error as a function of the number of training points. Since the test error is upper bounded by the training error, the absence of a noticable difference between the two curves when using the full training set indicates that we are indeed deriving a lot of power by simply having a lot of training examples.

We are also interested in obtaining some insight as to which features are more "influential" in training the models. This is assessed in figure 8.3 using two different metrics:

- Correlation with the output label: This is the "spearman" curve, which is simply the correlation between the given feature with the output labels. Scoring high on this is similar to what analysts typically associate with what constitutes a "powerful" variable: it should correlate strongly with the label and show visible sepration between them.

- Mutual information: This is the amount of information gained by observing the given feature [1]. More important should score high.

It is interesting to note that while the general trend between the two scores are similar, they are drastically different in certain features. It is not difficult to imagine a situation where the correlation scores low even

[1] Technically, it is the amount of information entropy that is lost.

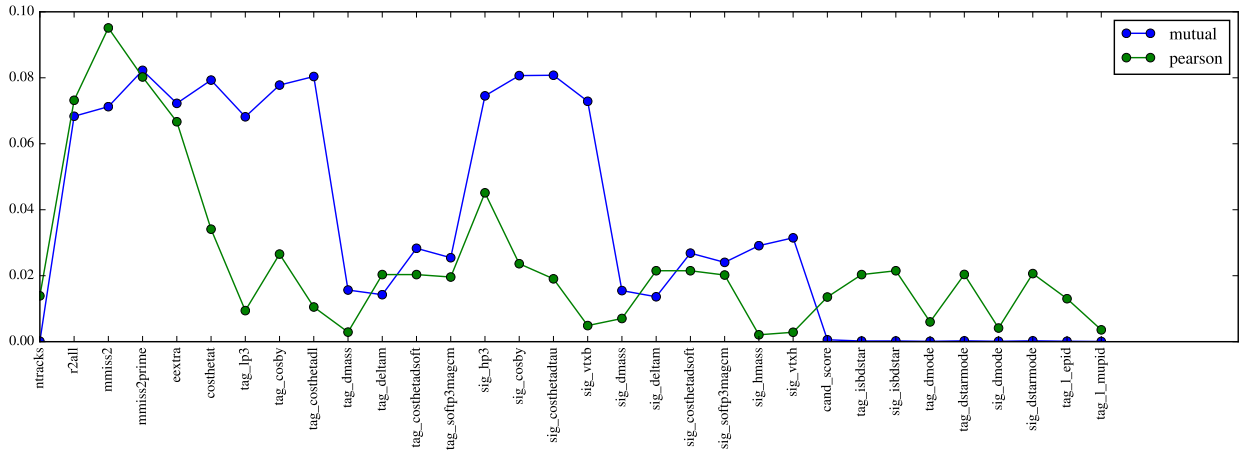


Figure 8.3: Relative importance of each feature for learning Z_1.

though it is very important. Take for example the problem of distinguishing two categories using just two features that happen to be Gaussian random vectors with the same mean, same variance, but with opposite correlation coefficients between the two features. Any single feature will show no correlation with the output, and indeed has no separation power on its own. However, the two taken together can be superb in distinguishing the categories.

8.1.3 Results

Figure 8.4 show the density function as well as data MC comparison for each of the models using the same plotting style as in chapter 7.

By visual inspection alone, the densities show that the three models are all somewhat similar. It can also be seen that the shapes between data and MC agree reasonably well while the normalization is possibly an effect due to branching fraction uncertainties. The random forest data MC comparison plot exhibits the unfortunate binning effect due to the limited precision discussed earlier, but it otherwise shows the same level of agreement as the others.

8.2 $D^*\tau$ detector

This learning problem is relevant to defining the statistic Z_2, which concerns distinguishing between the $D\tau$ and $D^*\tau$ signal event types.

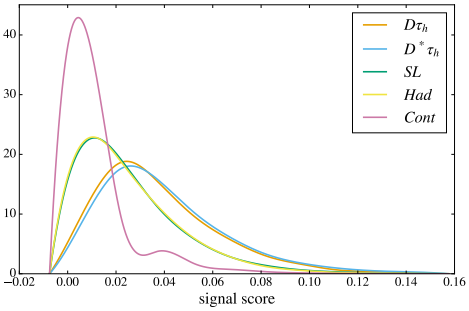
The training and testing data used both contain approximately $500K$ records, uniformly sampled from the signal MC, and consists of the ~ 34 features listed in chapter 7 plus the Z_1 score. The training labels are such that the $D\tau$ signal categories are labelled 1, and 0 otherwise.

8.2.1 Data preprocessing and learning models

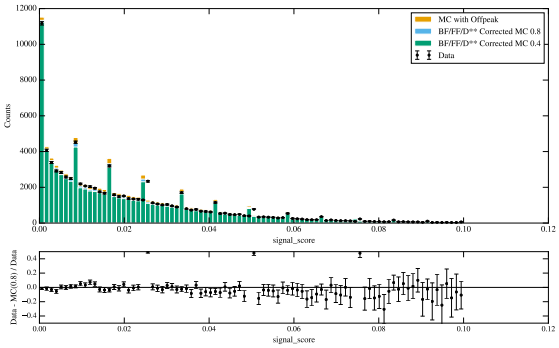
The data was preprocessed in the same way as in the signal detector, and the models attempted are also the same.

8.2.2 Analysis

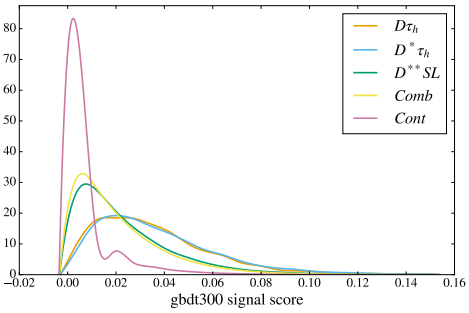
Figure 8.5 shows the ROC curves for each of the trained models. All of the models perform similarly, with the logistic regression trailing slightly behind.



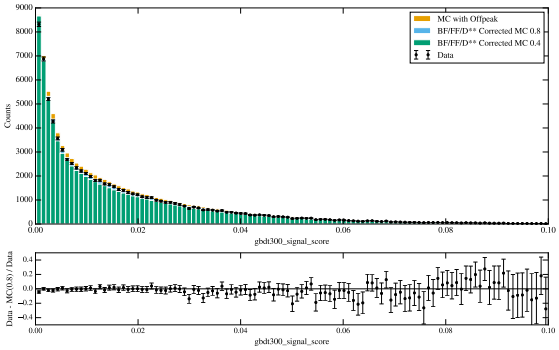
(a) Random forest event type densities.



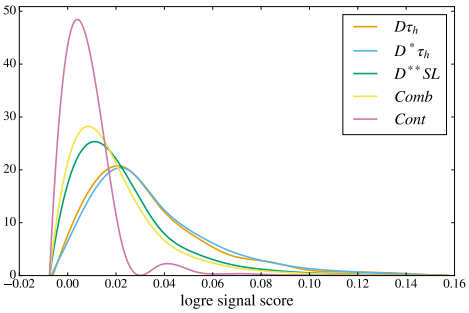
(b) Random forest data-simulation agreement.



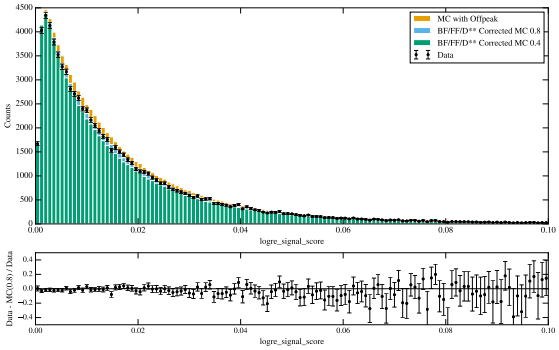
(c) Gradient boosting event type densities.



(d) Gradient boosting data-simulation agreement.



(e) Logistic regression event type densities.



(f) Logistic regression data-simulation agreement.

Figure 8.4: Density and data-MC agreement for the signal detectors.

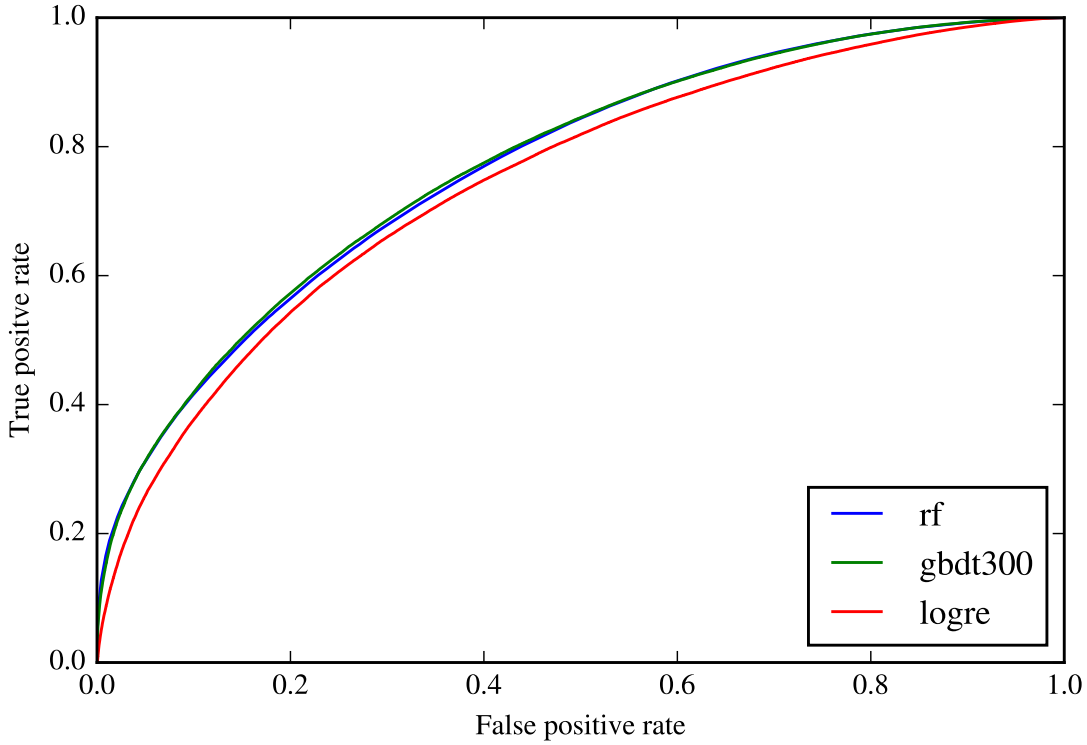


Figure 8.5: ROC curve for the Z_2 learners.

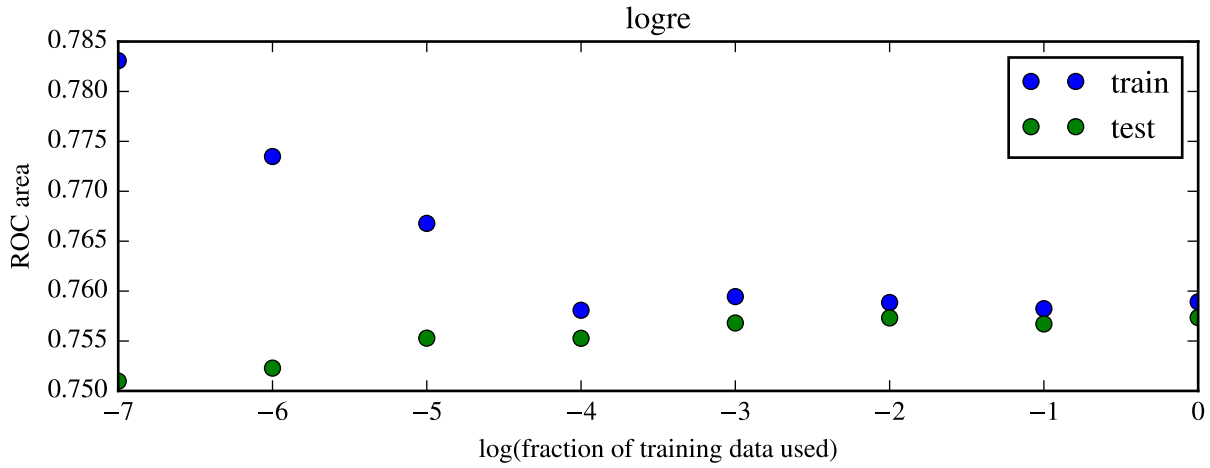


Figure 8.6: Learning curve for the logistic regression learner.

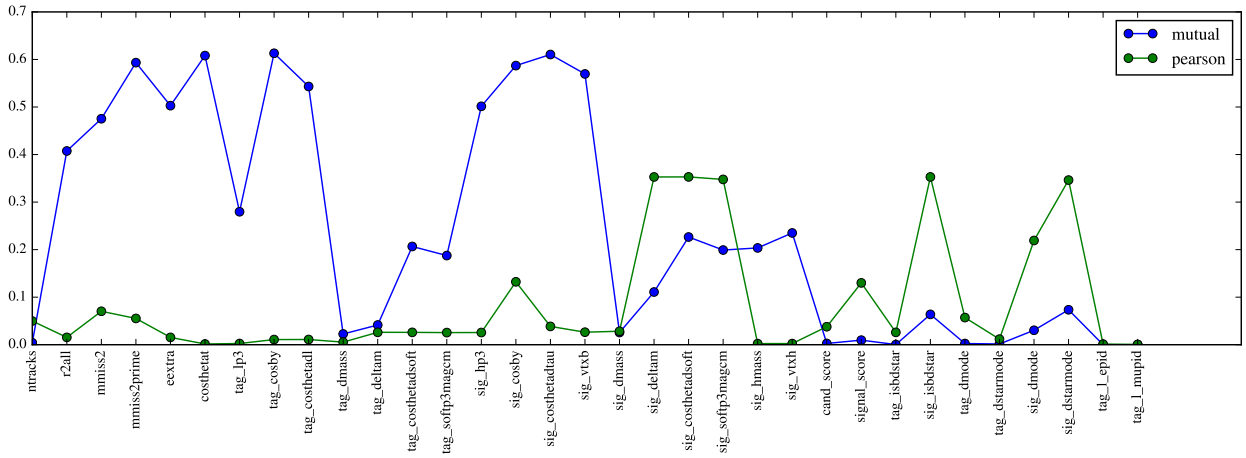


Figure 8.7: Relative importance of each feature for learning Z_2.

The learning curve for the logistic regression is shown in figure 8.6, which agains shows that it derives a lot of power by simply having a lot of data. In this case, logistic regression is doing slightly worse compared to the more complicated models. From this we conclude that the more sophisticated models really are finding special structure that a simple model couldn't.

Figure 8.7 shows the feature importances for this learning problem.

8.2.3 Results

Figure 8.8 show the density function as well as data MC comparison for each of the models using the same plotting style as in chapter 7.

The densities show that the three models do indeed find ways to separate the $D\tau$ and $D^*\tau$ categories, although the variation of the non-signal densities suggest that they seem to do so by focusing on different aspects of the dataset. It can also be seen that the shapes between data and MC agree reasonably well while the normalization is possibly an effect due to branching fraction uncertainties. The random forest data MC comparison plot exhibits the unfortunate binning effect due to the limited precision discussed earlier, but it otherwise shows the same level of agreement as the others.

8.3 Choice of Z_1 and Z_2

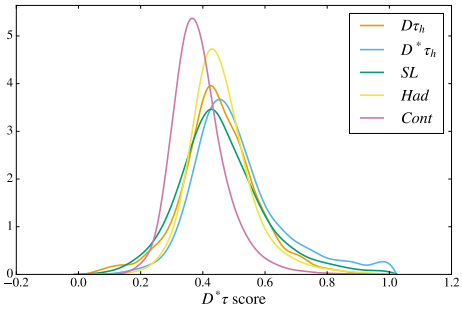
We use the logistic regression models as the basis for forming both Z_1 and Z_2. We could have based our decision purely on classification accuracy and chose gradient boosting trees for the $D^*\tau$ problem. However, we refrain from doing so since figure 8.8d suggests that it show signs of special structure that might eventually be more trouble than the marginal gain.

Let $f_1(X)$ and $f_2(X)$ be the output of the logistic regression models for the signal detector and $D^*\tau$ detector problem, repectively. We then define Z_1 and Z_2 as follows:

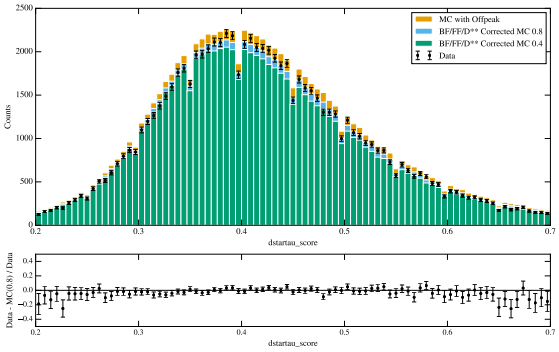
$$Z_1 = logit(f_1(X)) \tag{8.1}$$
$$Z_2 = logit(f_2(X)) \tag{8.2}$$

where

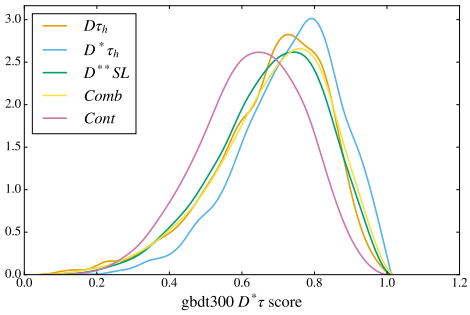
$$logit(x) = \log\left(\frac{x}{1-x}\right) \tag{8.3}$$



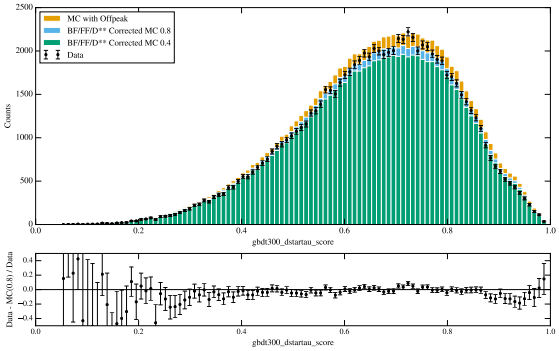
(a) Random forest event type densities.



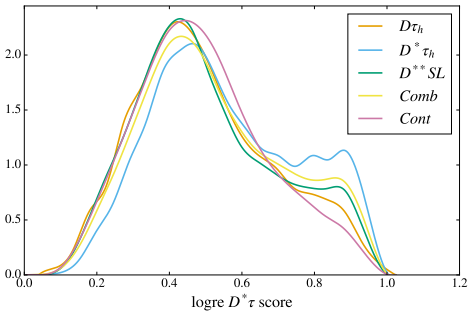
(b) Random forest data-simulation agreement.



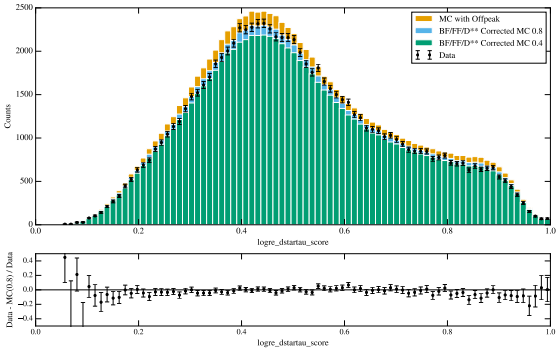
(c) Gradient boosting event type densities.



(d) Gradient boosting data-simulation agreement.

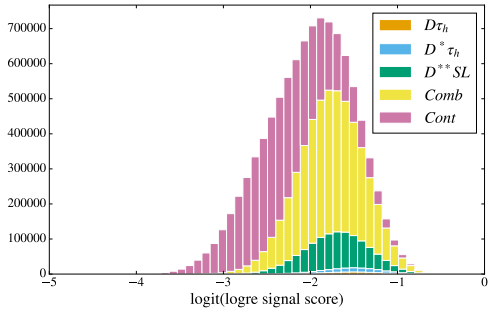


(e) Logistic regression event type densities.

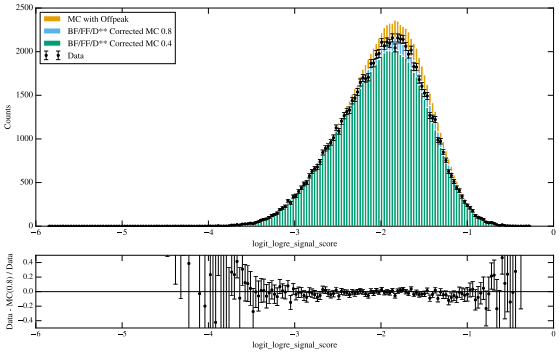


(f) Logistic regression data-simulation agreement.

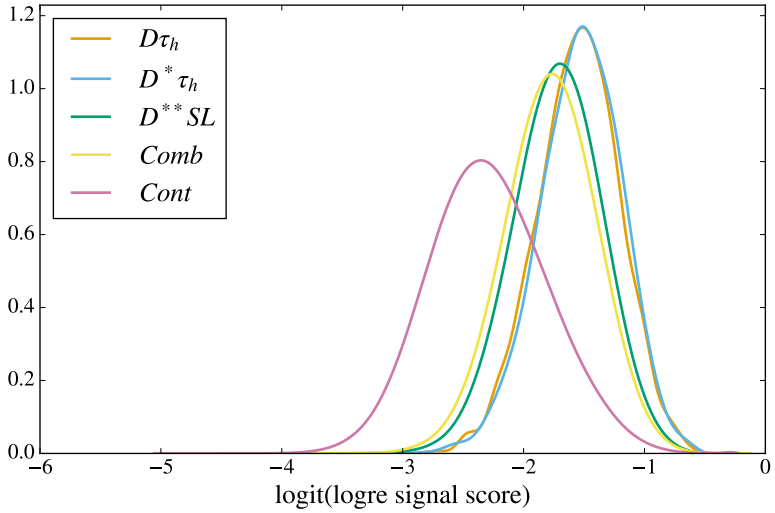
Figure 8.8: Density and data-MC agreement for the $D^*\tau$ detectors.



(a) Event type stacked histograms.



(b) Data-Simulation comparison.

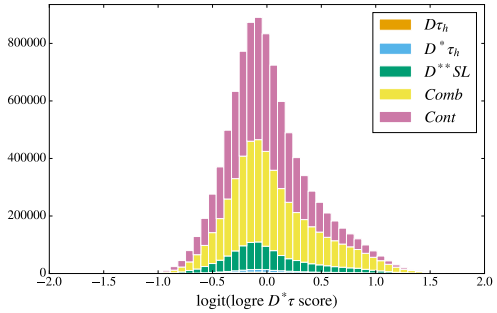


(c) Event type densities.

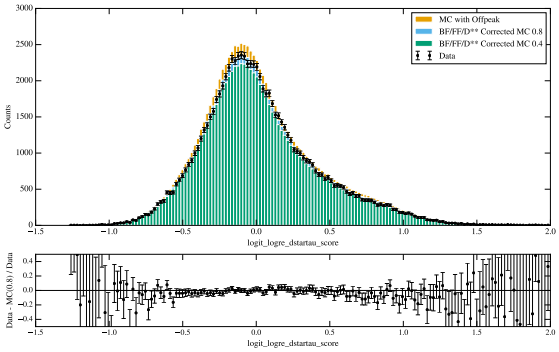
Figure 8.9: Z_1 score.

From a classification accuracy standpoint, applying these transformations are neutral since the *logit* is monotonic. However, their applications softens drastic changes in the distribution, thereby improving the quality of density estimation that we eventually obtain.

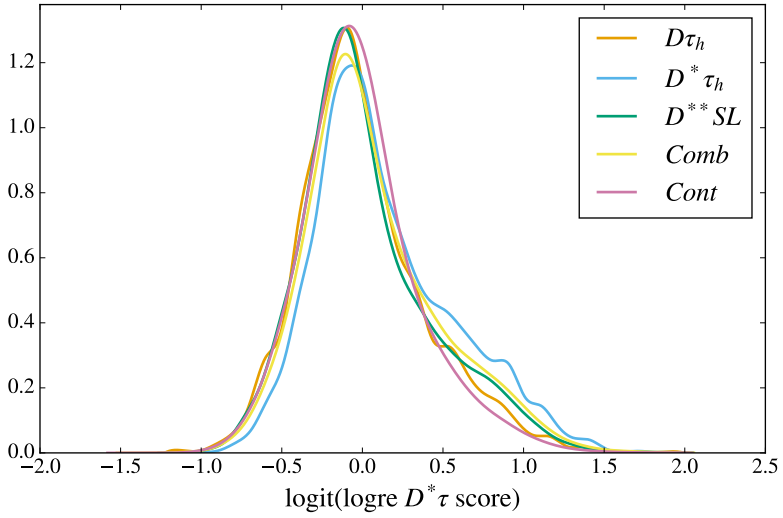
Figures 8.9 and 8.10 show the results for Z_1 and Z_2 respectively. The data and simulation agree reasonably well in their shapes, and the densities clearly show separation between the various event type categories. We also show the stacked event type histograms for reference.



(a) Event type stacked histograms.



(b) Data-Simulation comparison.



(c) Event type densities.

Figure 8.10: Z_2 score.

Chapter 9

Signal Extraction

This chapter discusses our method for estimating the proportion of the signal category. As discussed in chapter 4 this is equivalent to solving the optimization problem of equation 4.31, which we reproduce here for convenience:

$$\begin{aligned} \underset{p\in\mathbb{R}^{|C|}}{\text{minimize}} \quad & -\sum_{i=1}^{N} \log\left(\sum_{j\in C} p_j f_j(z_i)\right) \\ \text{subject to} \quad & \sum_{j\in C} p_j = 1 \end{aligned} \tag{9.1}$$

where

- C: The set of event type categories. They are $\{D\tau_h,\ D^*\tau_h,\ D^{**}SL,\ Comb,\ Cont\}$.
- N: Total number of On-Peak data records less those that are reserved for other purposes; e.g. data MC comparisons.
- z_i: Observed value of the signal detector score $Z = (Z_1, Z_2)$ for record i. These are precisely the two scores obtained by using the learners discussed in chapter 8.
- p_j: The optimization variable. It represents the proportion of records that belong to event category j.
- f_j: The (conditional) density function of Z for event type j.

Observe that the optimization problem 9.1 is *convex*, which means that we can reliably and efficiently find its global minimum by using a convex solver such as CVXOPT [19]. What remains is to fully specify the inputs[1]. Observe that the only items that remain unspecified in optimization problem 9.1 are the density functions for each event type. These are generally unknown, and we must model them in order to make progress.

For the moment, suppose that we had data sampled from a fixed event type. We would like to learn a corresponding density function of Z for that event type. There are several ways to proceed, but most of them fall in the following two categories:

- Parametric modelling: this approach assumes some parametrized functional form and uses the data to estimate those parameters. One of the shortcomings of this approach is that it is often impossible to know the exact functional form. Nonetheless, when applicable, this method is very effective and requires far less computing time compared to non-parametric approaches.

[1] Referece [30] has an excellent coverage on convex optimization and explains why their solution are essentially automatic. In fact, the hard part is really the ability to recognize and to frame the problem as a convex problem.

- Non-parametric modelling: this approach learns the density directly from the data without assuming any particular functional form. Its accuracy and correctness in the large sample limit is based on the law of large numbers[2]. The catch is that the computational complexity of the algorithms involved are often far larger than those in parametric statistics. This can often lead to unacceptable tradeoffs between running time and accuracy.

We will use a non-parametric approach known as the kernel density estimator to learn all of our event type densities. While it is true that this specific statistical methodology has been in frequent use in particle physics, we point out that the standard implementations are severely limited in terms of correctness. This lead us to implement an improved algorithm that solves not only the correctness issues, but also improves the running time drastically. The next section digresses from the physics analysis and focuses on how we perform kernel density estimates with an emphasis on the statistical and algorithmic issues and solutions.

9.1 Fast Kernel Density Estimation

The kernel density estimator[3] for the density function of a continuous random variable X over the dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ is the following:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{9.2}$$

where h is the *bandwidth* parameter that must be specified, and K is the kernel function which satisfies a few regularity conditions. Popular choices for K include Gaussian, Epanechnikov, and the sphere. We will sometimes refer to this dataset as the "training set" and denote it by $\mathcal{T}$.

To evaluate the kernel density estimator at point q, one can simply evaluate 9.2. Often, one is interested in evaluating over an entire set of queries, which we will refer to as the query set $\mathcal{Q} = \{q_i\}_{i=1}^N$. It is now easy to see the computational challenge involved: to evaluate a query set of size N will take time $O(N^2)$. This runtime complexity essentially renders the naive evaluation unusable when dealing with datasets involving millions of points[4].

It turns out that the choice of K does not affect the accuracy of the estimator very much; however, the choice of h is extremely important. We thus seek a criteria to to guide us in choosing the bandwidth. For this purpose, a common criteria defined for the purpose of assessing the quality of a kernel density estimator is to analyze its mean integrated squared error ($MISE$):

$$MISE \coloneqq \int (\hat{f} - f)^2 = \int \hat{f}^2 - 2\int \hat{f} f + \int f^2 \tag{9.3}$$

where f is the true density. The lower the $MISE$ the better the estimator. Observing that the last term of equation 9.3 is simply a constant, minimizing the following *cross validation* score over h is equivalent to minimizing the $MISE$:

$$CV(h) = \int \hat{f}^2 - 2N^{-1} \sum_i \hat{f}_{-i}(x_i) \tag{9.4}$$

where $\hat{f}_{-i}(x_i) = \frac{1}{Nh} \sum_{i \neq j} K\left(\frac{x_i - x_j}{h}\right)$; that is, simply evaluate 9.2 but withold the contribution due to point i itself. One of the attractive qualities of kernel density estimators is due to a theorem[31] which states that the bandwidth minimizing $CV(h)$ achieves the optimal $MISE$ in the large sample limit.

Therefore, to achieve good statistical performance, we must cross validate. However, it is immediately clear from equation 9.4 that naively evaluating the cross validation score for any single bandwidth h is $O(N^2)$. This is, again, not feasible for common people when N is in the millions.

[2] Many standard results, including those of density estimation, are grounded in the *Glivenko-Cantelli* theorem.

[3] For a more detailed exposition, see [20].

[4] Well, unusable for technology accessable to normal people. Then again, it is impossible for anyone if we go up just a few more factors.

One might wonder why many applications of kernel density estimation in particle physics appear to not run into any of the problems that we have discussed above. This is due largely to the following reasons:

1. Most commonly, evaluations of the kernel density is done only approximately. In fact, the algorithm most commonly used in particle physics[21] uses a greedy binning approach; indeed, this is what many other libraries outside of physics uses as well. Unfortunately, this algorithm has been shown to be very approximate, with potential for very large errors [32].

2. The dataset is just not large enough. While it is true that the raw size and number of data points collected at particle accelerators is unprecedented, the dataset that is actually derived for most analyses is tiny. Sometimes the raw record counts do not even exceed several tens of thousands, and its entirety can sometimes even fit in the memory of a reasonably powerful computer.

It turns out that there is, in fact, an efficient algorithm to evaluate both 9.2 and 9.4 that guarantees accuracy out to a specified precision in time $O(n)$ [22]. It is true that the hidden constants can be large and depends on the size of the bandwidth; however, we can improve this algorithm further by parallelizing the leaf computations on a graphics card [23]. The combination of these improvements led to a approximately a 10,000x speedup on problem sizes of a few million points; on our machines, the actual running time is just few seconds. We now turn to describing these two improvements and to presenting our performance benchmarks.

9.1.1 Dual tree evaluation

It is worth taking a moment to think about why the direct method for evaluating 9.2 and its naive improvements fall short in terms of correctness and runtime. The following are some of the most interesting observations that will lead to a more efficient algorithm:

1. The kernel function $K(x) \to 0$ as $x \to \infty$. In fact, the most efficient kernel has even a bounded support! The direct method does not take advantage of the fact that points far away from the query contributes little.

2. A naive way to take advantage of the distance information is to grid the data into aptly sized cells. To evaluate a query point, simply sum over contributions from points that are in the same cell as the query point. The shortcoming is obvious: if the query point is near the edge of the cell, it will simply ignore the contributions from the neighboring cell. The problem with this hack is that the fixed grid does not actually know where the data points are located.

Before writing down the dual-tree algorithm, we first consider the single-tree algorithm. This algorithm evaluates the kernel density estimate for a single point in time $O(\log n)$. It works by taking advantage of the finite extent of the kernel and by partitioning space adaptively into cells:

1. Use a space partition data structure such as the kd-tree [33] that automatically generates a grid that adapts to the spatial location of the training set. We now have a grid that is aware of where the training points are.

2. We can now take advantage of the limited extent of the kernel by *branch and bound*. The algorithm proceeds as if it had to evaluate the query point using every training point, but it knows when it is safe to neglect the contributions due to entire regions in feature space. The main stopping criteria are as follows:

   (a) If a region is sufficiently distant such that we can guarantee that collective contributions due to all the points within will not be more than a user specified tolerance, then we do not need to explore that region any further.

   (b) If all points within a region contribute sufficiently similarly such that the relative error is less than a user specified tolerance, then we do not need to explore that region any further.

Notice that all of the above approximations are guaranteed to meet the user's tolerance requirements. Of course, setting the error tolerance to 0 will just reduce to the direct approach, but this is essentially never the requirement since we are at least limited by the precision of floating point arithmetic. Actually, for kernels with bounded support, these "approxmiations" are actually not approximate at all since the contributions sufficiently far away is exactly 0.

Now that we can efficiently evaluate a single point, we can consider the problem of evaluating over an entire query set. We could just use the single-tree evaluation for each point in the query set; this approach will give a $O(n \log n)$ algorithm, and is already a massive improvement over the naive approach. However, we can do even better by partitioning the query set with a space partitioning data structure and by applying the same bounding tricks. This yields an $O(n)$ algorithm[5], which we show a sketch of in algorithm 2.

Algorithm 2 DUAL-TREE(Q, T)

Inputs:

- Q: Node of a kdtree built on the query set.
- T: Node of a kdtree built on the training set.

Procedure:

```
1: if CAN-APPROXIMATE(Q, T) then
2:     BOUND-CONTRIBUTIONS(Q, T)
3: else
4:     if Q and R are both leaf nodes then
5:         DUAL-TREE-BASE(Q, T)
6:     else
7:         DUAL-TREE(Q.left, T.left)
8:         DUAL-TREE(Q.left, T.right)
9:         DUAL-TREE(Q.right, T.left)
10:        DUAL-TREE(Q.right, T.right)
11:    end if
12: end if
```

Notes:

- Each kdtree node has *left* and *right* attributes which point to its left and right subtrees.
- CAN-APPROXIMATE is the function that decides whether points in Q are sufficently far from the points in T, and BOUND-CONTRIBUTIONS actually applies the bounding outlined in the main text.
- DUAL-TREE-BASE performs the direct evaluation of equation 9.2.

9.1.2 GPU acceleration

While it is true that the dual-tree algorithm already runs far more efficiently compared to the naive evaluation due to the improved computational complexity, the contants involved can still be large.

One can understand the running of algorithm 2 as two distinct phases:

1. Hierarchical tree traversal: this stage decides when it is possible to simply bound the contributions by points in a given spatial region.

[5]For a detailed discussion, see [32].

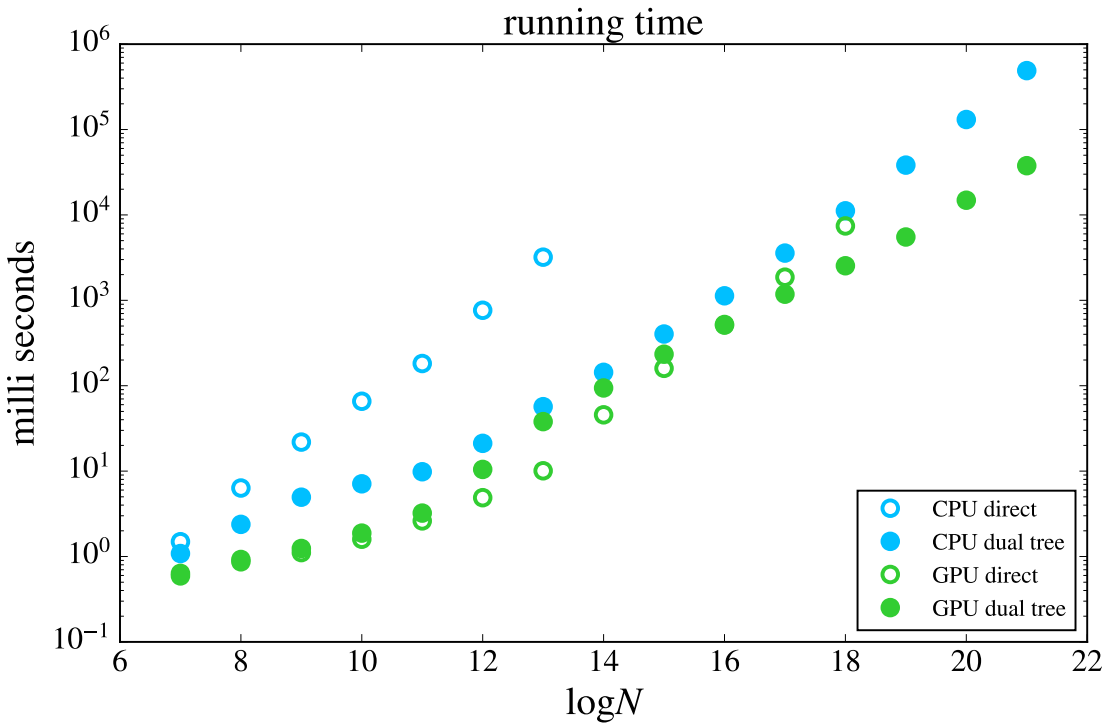


Figure 9.1: Runtime benchmark. N is the number of training and query points and *log* is in base 2. $1M$ is approximately when $\log N = 20$

2. Leaf evaluation: if the first phase can not justify making the approximations before reaching a leaf, it will be forced to perform the naive evaluation over the points residing at a leaf node.

Even though it is guaranteed that the number of leaf evaluations will be bounded such that the runtime complexity is $O(n)$, the hidden constants will strongly depend on the actual number of leaf evaluations, which can still be large when the extent of the kernel is large.

One way to speed this up is to observe that the direct evaluation lends itself very naturally to parallelization. In fact, the naive way of performing these kinds of N-body problems is one of the canonical ways in which GPU manufacturers measure and benchmark the improvements made between different chip generations. This lead us to algorithm 3 [23], which performs the direct evaluation on NVIDIA GPUs.

One might notice that algorithm 3 does not perform the parallelization strategy that equation 9.2 most directly suggests. That is, instead of creating $O(n^2)$ threads to parallelize each term in the computation and then finally doing a parallel aggregation, we instead assign each thread the task of aggregating all contributions for a single query point. This is due to the relative speeds of memory accesses on the GPU; it turns out that the computational efficiency that we gain in taking advantage of all the available parallism is overshadowed by the number of global memory accesses that it requires.

9.1.3 Results and benchmarks

Figure 9.1 benchmarks the runtime of each algorithmic improvement for kernel density estimation. The direct (dual-tree) algorithm is represented by an open (closed) circle, while the CPU (GPU) evaluation is represented using blue (green) circles.

We can clearly see the performance gain obtained by applying the various stages of algorithmic improvements. By extrapolating the runtime out to $\log N = 20$, we see that the total performance improvement

Algorithm 3 GPU-Dual-Tree-Base(Q, T)

Inputs:

- Q: Array of the query points.
- T: Array of the training points.

Parameters:

- N: Size of both Q and T. We have assumed that they are equal for simplicity.
- M: Maximum number of threads. Assume that M divides N.
- $t = 1, \cdots, M$: Thread index.

Procedure:

for all $i \in [1 \dots N/M]$ **do**
  Allocate array $A[M]$ on the GPU shared memory.
  Each thread loads the query point $q \leftarrow Q[(i-1)M + t]$ from GPU global memory.
  Each thread allocates its own accumulator $accum \leftarrow 0$ in a register.
  for all $j \in [1 \dots N/M]$ **do**
    parallel
    $A[t] \leftarrow T[(j-1)M + t]$
    syncthreads
    parallel
    for all $k \in [1 \dots M]$ **do**
      $accum = accum +$ Compute-Kernel(q,$A[k]$)
    end for
    syncthreads
  end for
  Each thread writes result back to GPU global memory $Q[(i-1)M + t].value \leftarrow accum$
end for

Notes:

- Q and T were tree nodes in algorthm 2, but they are now viewed as arrays of points.
- Each point in Q has an attribute *value* that stores its evaluation result.
- Compute-Kernel evaluates the kernel function. This is a single term in the summation of equation 9.2.
- CUDA has the concept of grids, blocks, and threads. The pseudocode here sweeps a lot of these details under the rug; it is, however, similar to having a single 1 dimensional block of M threads.

between the open blue circles (the direct CPU evaluation) to the closed green circles (dual-tree with GPU acceleration) is about x10,000.

9.1.4 Adaptive densities

One issue with 9.2 that can compromise the kernel density estimate's performance is that it uses a fixed bandwidth. It may be that certain areas of the feature space are far denser than others, so the optimal local bandwidths differ from those regions with lower densities. To allow for the possibilities of varying bandwidths over feature space, we can use the *adaptive kernel density estimator*[6]. It is defined according to the following procedure:

1. Use a *pilot estimate* $\tilde{f}(x)$, which can be the fixed bandwidth estimate of equation 9.2.
2. Define *local bandwidth factors* for each input data point x_i as follows:

$$\lambda_i = (\tilde{f}(x_i)/g)^{-\alpha} \tag{9.5}$$

   where g is the geometric mean over $\tilde{f}(x_i)$'s:

$$\log g = \frac{1}{N}\sum_{i=0}^{N} \log \tilde{f}(x_i) \tag{9.6}$$

   and $0 \leq \alpha \leq 1$ is a sensitivity parameter that decides the degree in which the local bandwidths should be sensitive to the local density. $\alpha = 0$ is equivalent to using the fixed bandwidth estimator.

3. Define the adaptive kernel density as follows:

$$\hat{f}(x) = \frac{1}{N}\sum_{i}^{H} \frac{1}{h\lambda_i} K\left(\frac{x - x_i}{h\lambda_i}\right) \tag{9.7}$$

   where K is the kernel function and h is an overall bandwidth.

The parameters h and α must be specified. It has been shown empirically that setting α to 0.5 is a good starting point, while opinions about setting h is more varied. One can of course choose these parameters by cross validation, but unlike the fixed bandwidth situation, there are fewer theoretical guanrantees regarding performance.

We also implement adaptive kernel densities; it turns out that it is relatively easy to generalize the branch and bound algorithm of 2 to accomodate for this case. Instead of deciding distance based on the overall bandwidth, decide it based on the largest bandwidth in a given cell.

9.1.5 Additional features

The following are some additional features implemented for our kernel density estimator. We will no go into further detail, and refer you to the project webpage.

- Can use arbitrary kernels. By default, we have implemented the Epanechnikov and Gaussian kernels.
- Arbitrary dimensions for CPU evaluation. 1D and 2D available for GPU, but can generalize to more.
- Support for weighted importance of the training points. This is especially useful for systematics that require re-weighting.
- Cross validation in 3-flavors: likelihood cross validation, least squares cross validation with convolution kernels, and least squares cross validation using numerical integration.

[6] See [20] for more details.

- Grid search for 2D training sets.
- Support for sampling from the estimated density.
- Marginal density projections.

9.2 Data samples for learning event type densities

The ideal data sample from which to learn the event type densities is to base it on real data that we know corresponds it *exactly*. This is clearly not achievable since it is in contradiction with the need for using our current analysis methodology; suppose we had some mechanism of isolating out a large and pure sample of the $D\tau$ category, then we could have just exploited this mechanism itself to extract the result.

Since it is impossible to achieve the ideal, we can try to learn the densities based on the following alternatives, each of which have their own strengths and weaknesses:

- Real data sufficiently similar to the specified event type: this is what is known as a "control sample". It is any sample of real data that is readily isolated, but also has the virtue of being similar to the the exact sample. It turns out that the presence of a high quality control sample is also not guaranteed. When this is the case, one should be especially careful about its role in the analysis and avoid unwarranted reliance simply because it is based on real data.
- Simulated data: the shortcoming for this is obvious. However, it has the virtue that it is possible to specify and to explictly state the input parameters and models based on our best knoweldge of physics. We can then assess how variations of that knowledge could affect the final result, leading to a more parametrized and controlled approach over the unknown.

The following lists the event types and the corresponding data sample from which we will learn the densities:

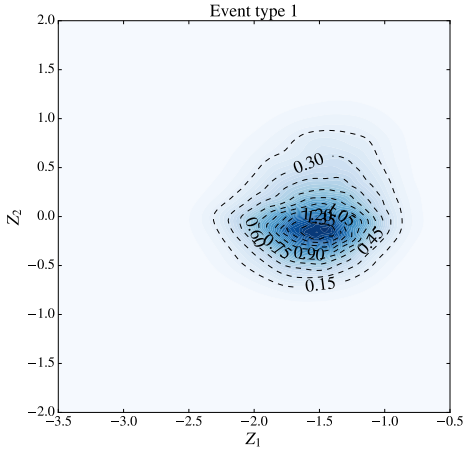
- $D\tau$: Generic MC.
- $D^*\tau$: Generic MC.
- $D^{**}SL$: Generic MC.
- $Comb$: Generic MC. We will use a sideband control sample to assess what the result could have been assuming it were a good representation of the exact sample.
- $Cont$: Off-Peak data. We consider this to be a good control sample

9.3 Estimated densities

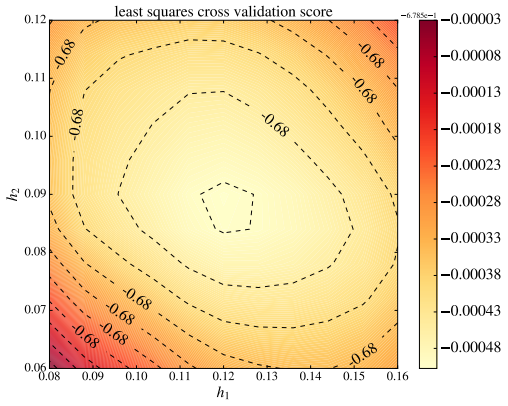
Figures 9.2, 9.3, 9.4, 9.5, and 9.6 show the estimated densities for each event type. Each figure contains 4 subfigures ordered from left to right, and then top to bottom. They show the following:

1. The 2D density that was estimated. This is what we will substitute for the f_j's into the optimization problem 9.1.
2. The least square cross validation score was evaluated over a grid for the purpose of finding the optimal bandwidth. In particular, we simply use the local minimum found here for the optimal bandwidth. These are then used for the bandwidths for 2D densities shown.
3. The 1D marginal density in Z_1. A histogram of the training points are also shown; this is done as a cross check that the learned density is behaving as expected.
4. The 1D marginal in Z_2.

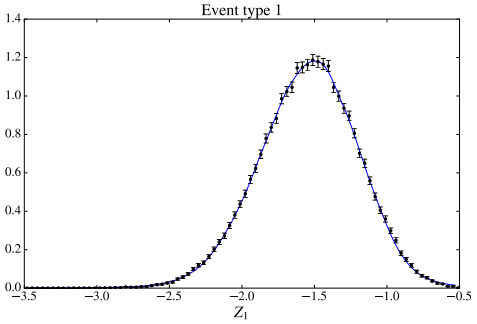
The estimated densities all appear to behave as expected.



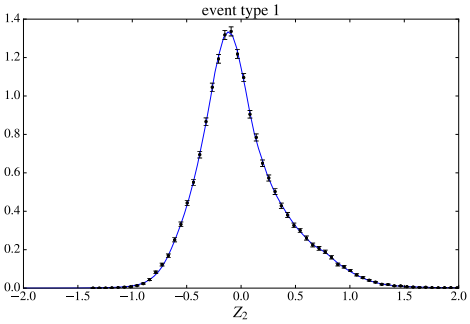
(a) Estimated 2D density.



(b) Least square cross validation scores.

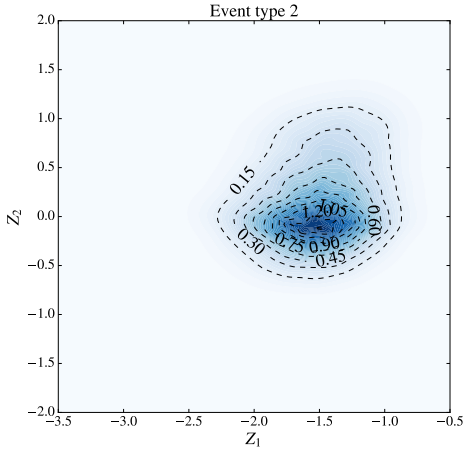


(c) Marginal density in Z_1.

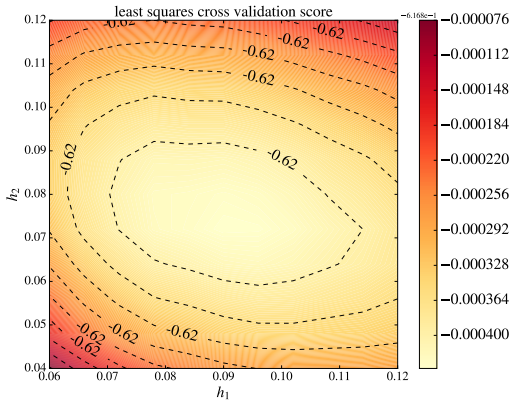


(d) Marginal density in Z_2.

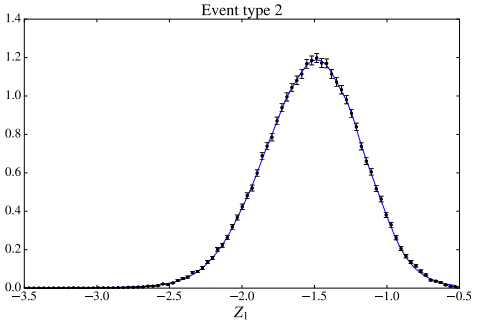
Figure 9.2: Kernel density estimates for the $D\tau$ event type.



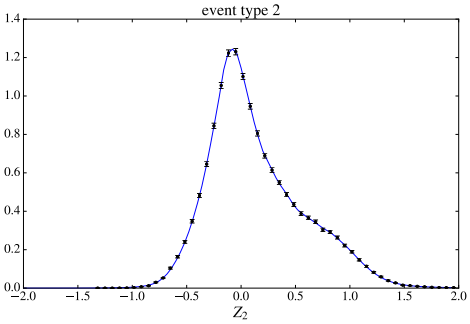
(a) Estimated 2D density.



(b) Least square cross validation scores.

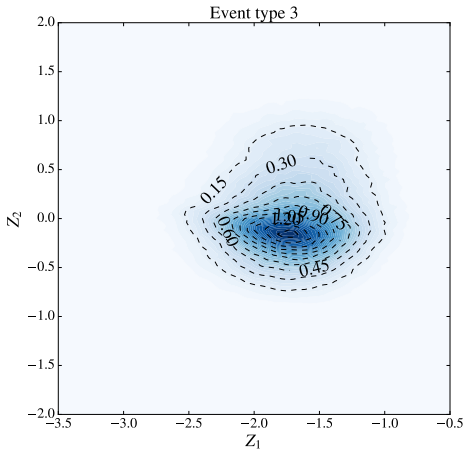


(c) Marginal density in Z_1.

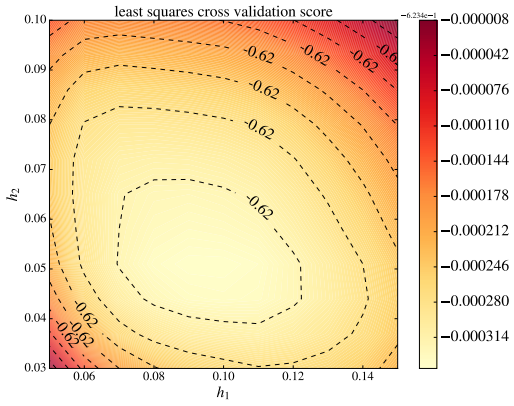


(d) Marginal density in Z_2.

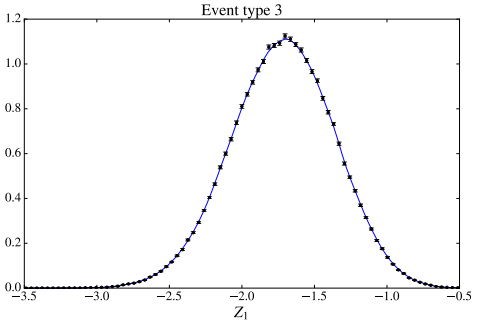
Figure 9.3: Kernel density estimates for the $D\tau$ event type.



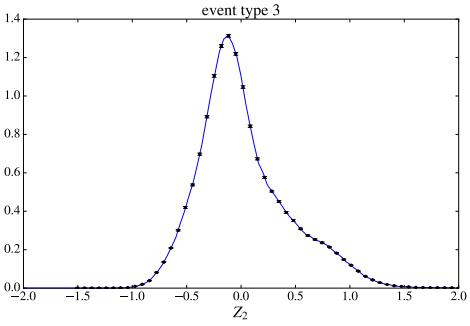
(a) Estimated 2D density.



(b) Least square cross validation scores.

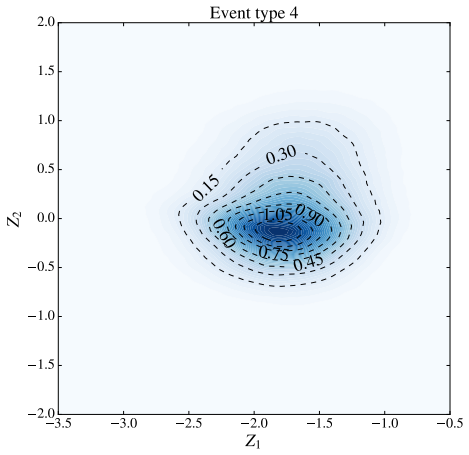


(c) Marginal density in Z_1.

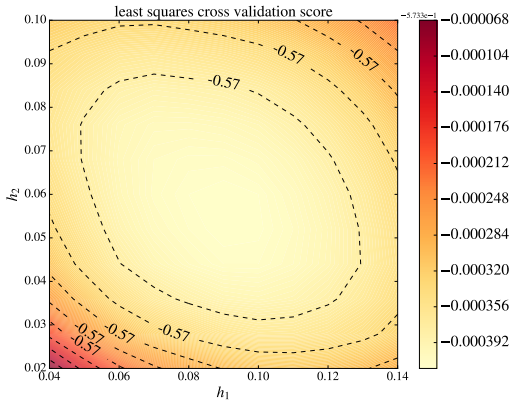


(d) Marginal density in Z_2.

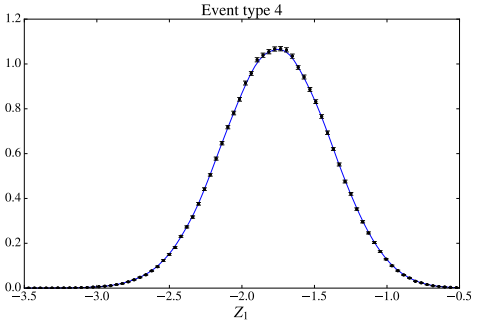
Figure 9.4: Kernel density estimates for the $D\tau$ event type.



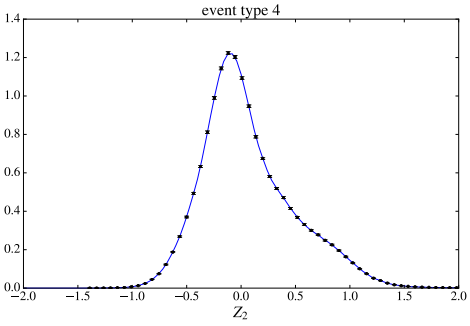
(a) Estimated 2D density.



(b) Least square cross validation scores.

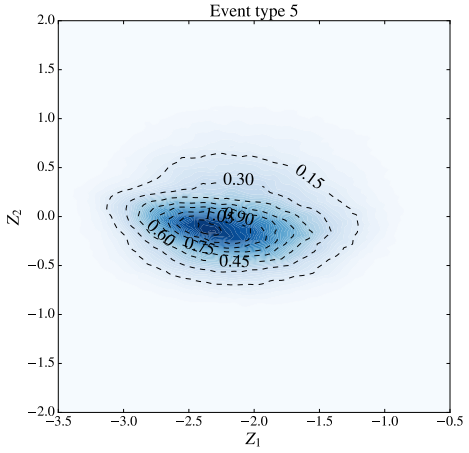


(c) Marginal density in Z_1.

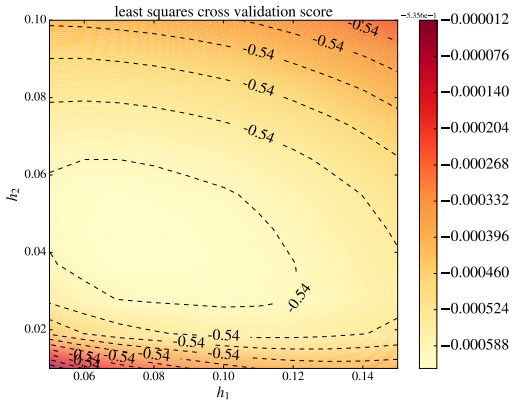


(d) Marginal density in Z_2.

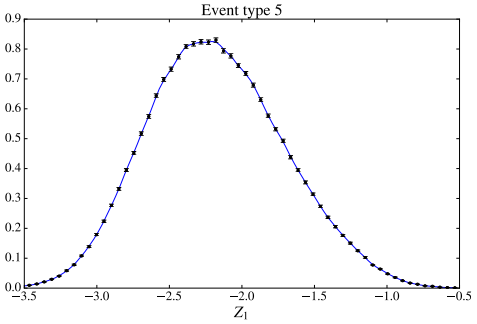
Figure 9.5: Kernel density estimates for the $D\tau$ event type.



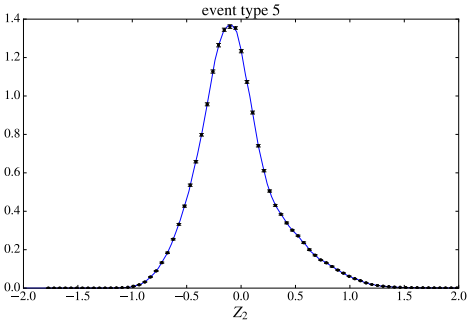
(a) Estimated 2D density.



(b) Least square cross validation scores.



(c) Marginal density in Z_1.



(d) Marginal density in Z_2.

Figure 9.6: Kernel density estimates for the $D\tau$ event type.

9.4 Bias Correction of Extracted Signal Proportions

This chapter has, thus far, only prescribed a procedure to obtain the estimator $\hat{p}_i$ for each p_i. In practice, these estimators can have large biases that must be corrected or be accounted for directly as a systematic uncertainty. Our analysis takes the approach to correct for them, and we outline the procedure in this section. Since the measurement is ultimately regarding the signal components, we only need to be concerned with the biases of $p_{\hat{D}\tau}$ and $p_{\hat{D^*}\tau}$.

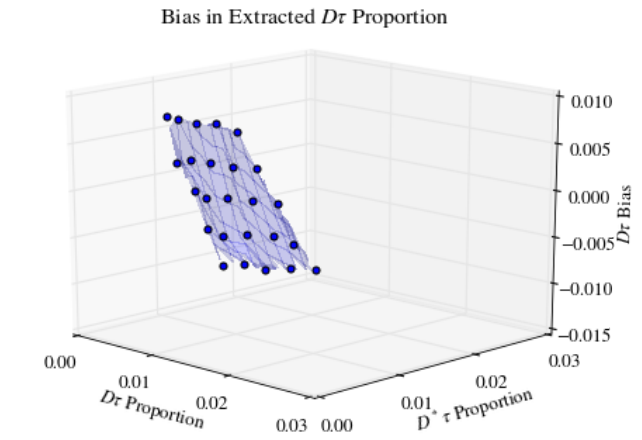
The approach we take is to approximate two mappings. The first maps the pair $(p_{\hat{D}\tau}, p_{\hat{D^*}\tau})$ to $b_{D\tau}$, and the second maps the pair to $b_{D^*\tau}$, where $b_{D^{(*)}\tau}$ is the bias of the estimator $p_{\hat{D^{(*)}}\tau}$. For concreteness, we focus our discussion on approximating the first mapping, which we denote T. The procedure for the second mapping proceeds analogously.

We will approximate T by estimating its value at a set of grid points, and interpolate between them as needed during the analysis. We will frequently refer to this set of evaluations of T as the "bias lookup table".

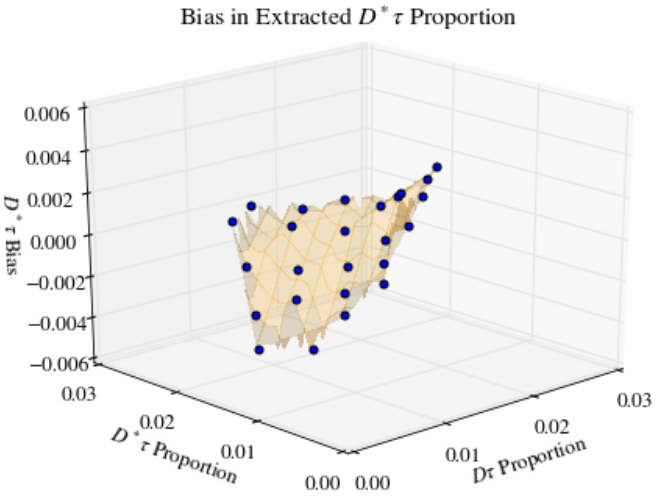
Before describing how we might obtain a single entry of this table, we remind the reader of two mutually exclusive samples of simulated data that we have pre-allocated. The first is the sample in which the KDE's are learned, and the second a hold out ("test") set.

To obtain one entry, pick a pair of values $(p_{D\tau}, p_{D^*\tau})$. We resample the test set 40 times, with replacement, such that each sample containes exactly $p_{D^{(*)}\tau}$ worth of the $D^{(*)}\tau$ component while scaling the others components accordingly. For each sample, we solve 9.1. Once all of the 40 $p_{\hat{D}\tau}$ are computed, we average them and less the input $p_{D\tau}$ to estimate the bias. The pair of averages of $(p_{\hat{D}\tau}, p_{\hat{D^*}\tau})$ and the corresponding bias value will then serve as an entry in this lookup table. After using 45 pairs of initial $(p_{D\tau}, p_{D^*\tau})$ pairs, we obtain the bias table shown in Figure 9.7.

The look-up table as described above assumes that the biases only depend on the differences of signal proportions. The effects of the differences between the background proportions are explored as a systematic uncertainty in Chapter 11.



(a) Bias in extracted $D\tau$ proportion.



(b) Bias in extracted $D^*\tau$ proportion.

Figure 9.7: Biases of the signal proportions as a function of the extracted signal proportions. The points are linearly interpolated to better showcase the overall shape.

Chapter 10

Solving for $\hat{p}_j$ and $\hat{\epsilon_j}$ on simulated data

This chapter solves for the $\hat{p}_j$'s by solving the optimization problem 9.1 on simulated data. The goal of this is mainly two fold. The first is to check that the procedure is self consistent and exhibits the statistical properties that are expected of the algorithms used; in other words, this is for debugging. To make this toy example easier to work with, we have fine tuned the bandwidths manually so that the estimated values on the simulated data is unbiased.

We will also derive the values for $\hat{\epsilon}_j$ and compute its statistical uncertainties. These will be used as-is when we derive the final result from the detector data.

10.1 Solving for $\hat{p}_j$

To obtain the central value as well as the standard error for $\hat{p}_j$, we bootstrap the data sample 1200 times and solve the optimization problem of 9.1 on each of these instances.

The results are shown in figure 10.1. We note that the sampling distribution for the $\hat{p}_j$'s are approximately Gaussian, as we would have expected of maximum likelihood estimates.

Figure 10.2 show the marginal distributions when the central values of the $\hat{p}_j$'s are substituted for the joint density of Z_1 and Z_2. The results look consistent on visual inspection.

The results in figure 10.1 show that the statistical errors for the $D\tau$ and $D^*\tau$ categories, are respectively, $\boxed{0.0007}$ and $\boxed{0.0005}$.

Looking up the bias correction as described in Section 9.4 leads us to make correction of -0.0025 and -0.0063 for the $\hat{p}_{D\tau}$ and $\hat{p}_{D^*\tau}$, respectively. When bias-corrected, the extracted proportions agree well with the known values of the signal proportions.

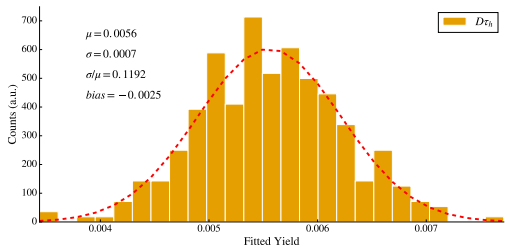
10.2 Solving for $\hat{\epsilon}_j$

For each signal category j, we estimate the efficiency as follows:

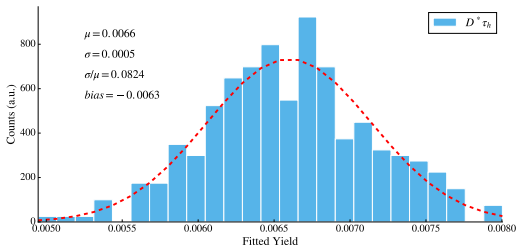
$$\hat{\epsilon}_j = \frac{\sum_{i=1}^{N^{(j)}} w_i}{\sum_{i=1}^{M^{(j)}} w_i} \tag{10.1}$$

where we have defined the following quanitites:

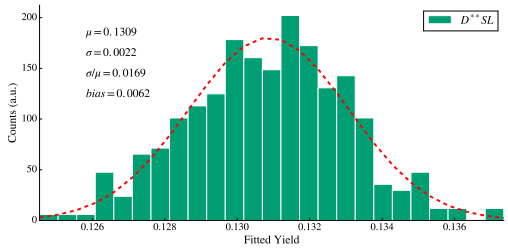
- $N^{(j)}$: number of records belonging to category j that is in the sample that has passed all of our data filtering criteria.
- $M^{(j)}$: number of records belonging to category j that was generated in the simulation.



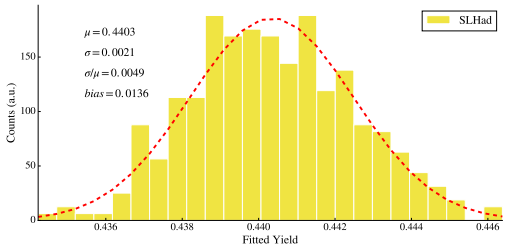
(a) $D\tau$.



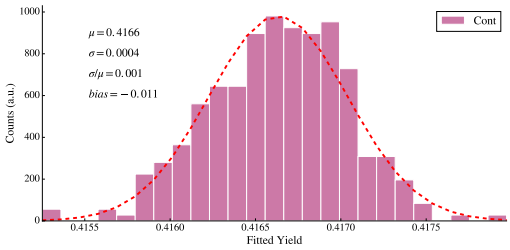
(b) $D^*\tau$.



(c) $D^{**}SL$.



(d) *Comb*.

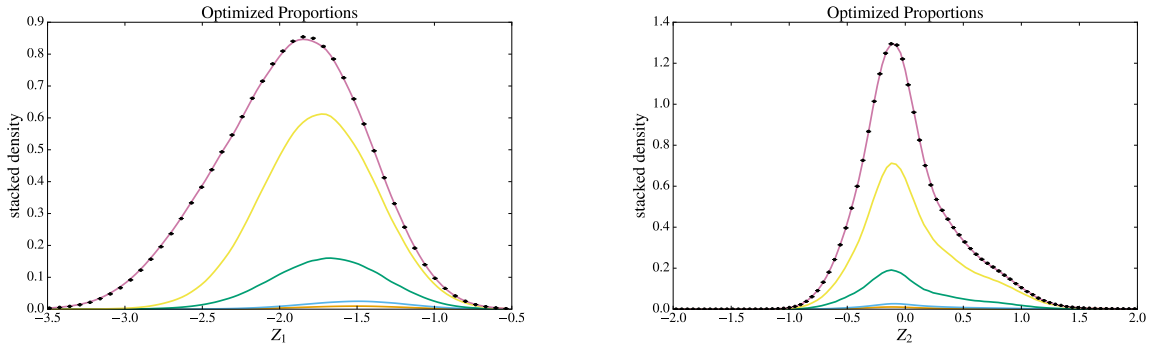


(e) *Cont*.

Figure 10.1: Bootstrapped results

- w_i: the weight of the record.

Applying the formula above, we obtain efficiencies of $\boxed{1.29\%}$ and $\boxed{1.06\%}$, for the $D\tau$ and $D^*\tau$ categories, respectively. The statistical errors are negligible.



(a) Z_1. (b) Z_2.

Figure 10.2: Stacked event types.

Chapter 11

Systematic Uncertainties

11.1 Overview

The systematic uncertainties associated with this analysis are primarily due to our dependency on the simulated data. In particular, the following steps rely on the efficacy of the simulation to represent reality:

1. To estimate p_j in equation 4.27, we solved the optimization problem of equation 4.31. This optimization problem required simulation input to specify the distribution shapes $\hat{f}_j$; hence, the uncertainty in the simulation manifests as an uncertainty in the estimated distribution $\hat{f}_j$.

2. To estimate ϵ_j in equation 4.27, we used simulated data to evaluate each term in equation 4.40.

This chapter will discuss, in sequence, factors that affect the simulation. For each factor, we discuss how it affects each item mentioned in the list above and quantify the uncertainty it introduces into the final result. The last section will summarize the results derived in this chapter and collect them into table 11.14.

We begin by defining a numeric quantity that we will be using to quantify any given systematic uncertainty. First, denote the actual physical model that generated the detector data to be β. Recall that we also have some reference physical model, encoded in simulation, that we will use to obtain our final result by fitting it to the data. We use α to denote this "central" model. Consider a systematic variation away from this "central" model that captures our ignorance. We us α' to denote this model.

We will denote the bias of $\hat{p}$ when using model α to fit to the data generated from model β to be $b^{(\alpha,\beta)}$. It is by definition,

$$b^{(\alpha,\beta)} = \mathbb{E}\left[\hat{p}^{(\alpha,\beta)}\right] - p^{(\beta)} \tag{11.1}$$

We use the notation $\hat{p}^{(\alpha,\beta)}$ to make it clear that the estimator is obtained when fitting β with model α, and use the notation $p^{(\beta)}$ to emphasize that the true proportions are those in model β. Quantities that denote the case when we use model α' to fit to model β are defined analogously.

We make the assumption[1] that the $b^{(\alpha,\beta)}$ can be decomposed as

$$b^{(\alpha,\beta)} = b^{(\alpha)}_{KDE} + b^{(\alpha,\beta)}_{model} + b_{other} \tag{11.2}$$

where

- $b^{(\alpha)}_{KDE}$ is the bias contributed purely from the KDE's failure to represent model α.
- $b^{(\alpha,\beta)}_{model}$ is the bias contributed purely from the fact that model α differs from model β. In particular, $b^{(\alpha,\alpha)}_{model} = 0$, and similarly for α'.

[1] One can certainly challenge this, but we observe some empirical evidence that suggest that this is not so bad.

- b_{other} is the remaining bias, which we assume to be negligible. This will included terms such as the MLE's inherent bias, which we assume to be small for large sample sizes. We will be omitting this term in what follows.

We define the systematic uncertainty due to the variation α' from α when fitting to data β to be

$$\Delta^{(\alpha',\alpha,\beta)} = b^{(\alpha',\beta)}_{model} - b^{(\alpha,\beta)}_{model} \tag{11.3}$$

Observing that $b^{(\alpha,\alpha)} = b^{(\alpha)}_{KDE}$, and analgously for α', we can rewrite equation 11.3 as

$$\Delta^{(\alpha',\alpha,\beta)} = \left(\mathbb{E}\left[\hat{p}^{(\alpha',\beta)}\right] - \mathbb{E}\left[\hat{p}^{(\alpha,\beta)}\right]\right) - \left(\mathbb{E}\left[\hat{p}^{(\alpha',\alpha')}\right] - \mathbb{E}\left[\hat{p}^{(\alpha,\alpha)}\right]\right) + (p^{(\alpha')} - p^{(\alpha)}) \tag{11.4}$$

We obtain the first quantity on the RHS by taking the difference between the estimator values for when using model α' vs using model α to fit to the real data. To obtain the second quantity, we simply using model $\alpha^{(')}$ to fit to data generated using model $\alpha^{(')}$. The third quantity is known since both α and α' are models dialed into the simulation.

All of the systematic uncertainties in the following sections will be obtained by computing the quantity defined in equation 11.4. Note that the model β in the forgoing discussion is the actual model underlying the true physical mechanism, but in what follows, we also quote systematic uncertainties for when β represents the model of data generated from simulation. The purpose of the latter is for debugging and done before we were unblinded.

11.2 Form factors

Whether our estimated density functions accurately represents those that would have been observed under the Standard Model depends on how well we can simulate its dynamics. The difficulty of this task is almost entirely due to our general lack of knowledge for how QCD behaves at low energies. Since this problem is manifest in all levels of the simulation, we focus only on what we believe are the most consequential aspects.

Specifically, we explore how different hadronization models can influence the semileptonic B decay simulation, which indirectly influence the estimated density functions $\hat{f}_j$ and hence $\hat{p}_j$. We do not specifically consider their impact on $\hat{\epsilon}_j$, since such variations only manifest through variations in detector efficiencies, for which we believe have already been conservatively upper bounded in section 11.5.

Obviously, it is not practical to simulate an entirely new set of data for each model that we care to explore; instead, we apply a set of weights to each record of the simulated data that was generated based on the default settings. Suppose we would like to assess the impact of using model A. Denoting the default model by O, we derive a weight for an arbitrary record in the default simulated dataset as follows:

1. Identify a set of variables x, often kinematic in nature (e.g. q^2, E_ℓ, etc.), whose possible variations in its distribution are considered important when changing to the model A.

2. Let $d\Gamma_A/dx$ be the differential decay rate under model A and define $\Gamma_A = \int d\Gamma_A/dx\, dx$. Similarly, define $d\Gamma_O/dx$ and Γ_O.

3. For the given record, compute x and its differential decay rates.

4. Define the weight for this record to be

$$w(x) = \frac{\frac{1}{\Gamma_A}\frac{d\Gamma_A}{dx}(x)}{\frac{1}{\Gamma_O}\frac{d\Gamma_O}{dx}(x)} \tag{11.5}$$

5. Use the weighted data points to re-estimate the densities and to extract the results.

This procedure is widespread in physics analyses. The only improvement that I made in this procedure is in the computation of Γ; people have, in the past, used surprisingly cumbersome methods to compute the integral. The code base has seen everything from Monte Carlo integration to integrating hard coded polynomial approximations. While such methods are useful and sometimes required in certain situations, they are not necessary in this use case; in fact, they are less accurate and are far more prone to numerical errors than simply using a deterministic integration algorithm. Therefore, all we did is reduce this to a one liner by making a call to a well known gaussian quadrature code[2].

In the following sections, we discuss each form factor model that were considered and quantify their impact as systematic uncertainties. We will cite and credit the original authors of these models in the appropriate sections, but I take the moment now to specifically mention work done by Michael Mazur and Jeff Richman (BAD 1111) and work done by Kenji Hamano and Bob Kowalewski (BAD 1586), who have consolidated the garden variety of form factor parameterizations into a handful that are directly useful in performing the re-weighting. We also credit Art Snyder, whose kinematic computing code is borrowed by many others.

11.2.1 Uncertainties due to $B \to D^{(*)}\ell\nu_\ell$ form factors

We consider the impact of form factor models on the following kinematic variables for semileptonic B decays to $D^{(*)}$ mesons:

- q^2: Momentum transfer from the B meson to the virtual W boson. This sets the energy scale for which hadronization of the $D^{(*)}$ meson takes place; dynamics at higher q^2 are better understood than those at lower values.
- θ_ℓ: Angle of the three momentum of the ℓ in the virtual W's rest frame.
- θ_V: Angle of the three momentum of the D in the D^*'s rest frame. Applicable only to $B \to D^*$ decays.
- χ: Angle between the decay planes of the W and the D^*. Applicable only to $B \to D^*$ decays.

Given a specific form factor model, we compute the differential decay rates in these kinematic variables using the formulas provided by Korner and Schuler[9]. It remains to identify and specify the form factor model settings that are used in this analysis. They are as follows:

- Settings used to generate the simulated data:
  - $B \to D\ell\nu$, $\ell = e, \mu, \tau$: ISGW2[34].
  - $B \to D^*\ell\nu$, $\ell = e, \mu$: Linear q^2[35]. The parameter settings are
    - $\rho^2 = 0.77$
    - $R_1 = 1.33$
    - $R_2 = 0.92$
  - $B \to D^*\tau\nu$: ISGW2.
- Settings re-weighted to in obtaining the central value:
  - $B \to D\ell\nu$, $\ell = e, \mu, \tau$: CLN[10]. The parameter settings are
    - $\rho^2 = 1.186$
    - $V_1 = 1.0816$
    - $\Delta = 1.0$
  - $B \to D^*\ell\nu$, $\ell = e, \mu, \tau$: CLN[10]. The parameter settings are

[2]This is the same code used in the GNU scientific library as well as those in MATLAB. It is by Pavel Hodolborodko. See `http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/`

    * $F_1 = 0.921$
    * $\rho^2 = 1.207$
    * $R_0 = 1.14$
    * $R_1 = 1.401$
    * $R_2 = 0.854$

- Settings re-weighted to in obtaining the systematic uncertainty:
  - $B \to D\ell\nu$, $\ell = e, \mu, \tau$: CLN[10]. The parameter settings are
    * $\rho^2 = 1.186 \pm 0.054$
    * $V_1 = 1.0816$
    * $\Delta = 1.0$
  - $B \to D^*\ell\nu$, $\ell = e, \mu, \tau$: CLN[35]. The parameter settings are[3]
    * $F_1 = 0.921$
    * $\rho^2 = 1.207 \pm 0.026$
    * $R_0 = 1.14$
    * $R_1 = 1.401 \pm 0.033$
    * $R_2 = 0.854 \pm 0.02$

Figures 11.1 and 11.2 show the generated simulated data overlayed with the form factor settings that they are generated from. The agreement is quite good as we might have expected, except for the light leptons at low q^2. This effect is due to the failure of the plotted model curve to account for the mass widths of the B and D mesons. When the generated mass values are fixed to their central values, the agreement becomes effectively perfect.

Figures 11.3 and 11.4 show the different form factor models that are considered for $B \to D^{(*)}$ decays. Notice that ISGW2 is quite different from those that are derived from HQET; even within HQET, different assumptions applied within the QCD sum rules still lead to slight differences.

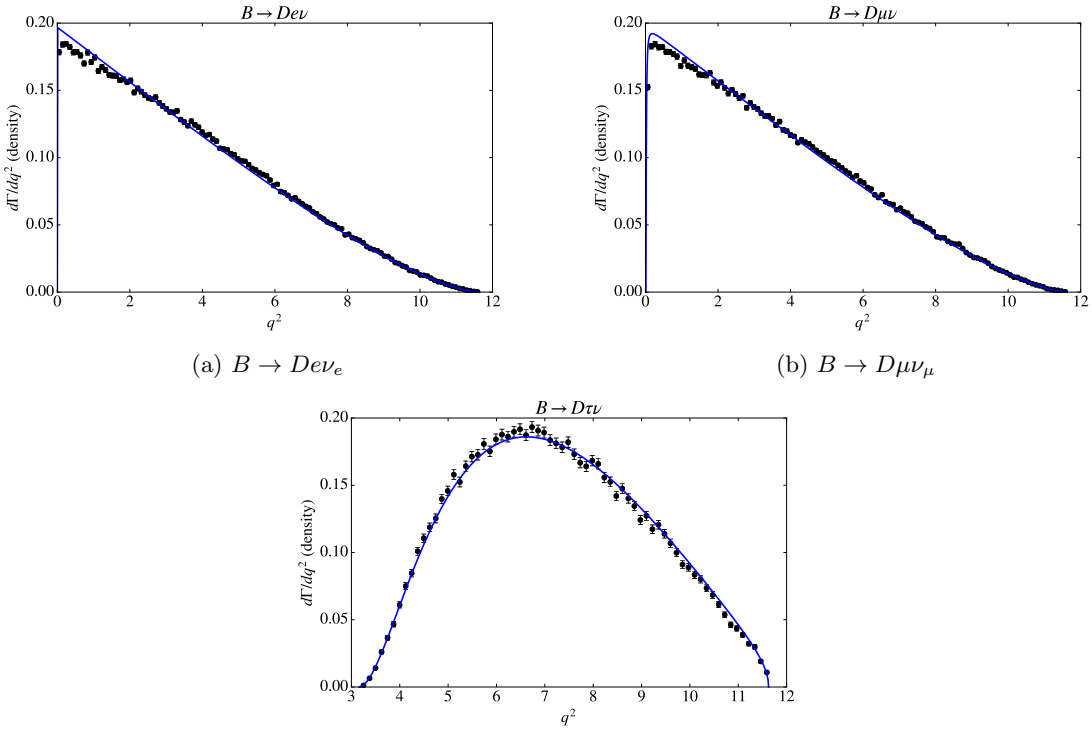
Figures 11.5 and 11.6 show the q^2 spectra for the variations of the $B \to D\ell\nu$ and $B \to D^*\ell\nu$ CLN parameters, respectively. The effects of the slight change in the shapes are assessed as the form factor systematics.

To obtain the central value, we re-weight the default simulated points to the "central value" settings specified above. These weights are then used as input to learn the densities, which are then used as input to extract the (bias-corrected) result from detector data. To obtain the systematic uncertainty, we extract the result in the same way except that we re-weight to the "systematic" settings above. The difference between the value obtained this way and the central value is then listed as the systematic. The results are shown in table 11.1. As a cross check, we also perform this procedure on the simulated data.

Signal Type	Simulated data	Detector data
$D\tau$	0.0005	0.0030
$D^*\tau$	0.0014	0.0024

Table 11.1: Systematic uncertainties on $\hat{p}_j$ due to $B \to D^{(*)}$ form factors' influence on $\hat{f}_j$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

[3]Note that the variations of the $B \to D^*\ell\nu$ form factors were along the principal axis from taking the correlations into account.

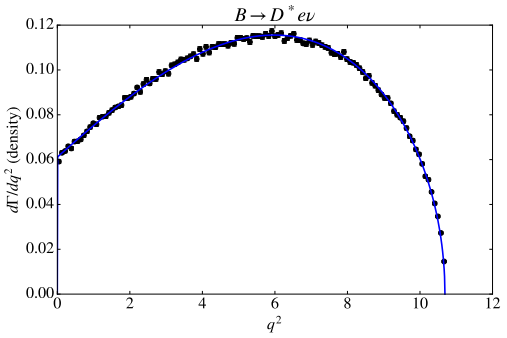


(a) $B \to De\nu_e$

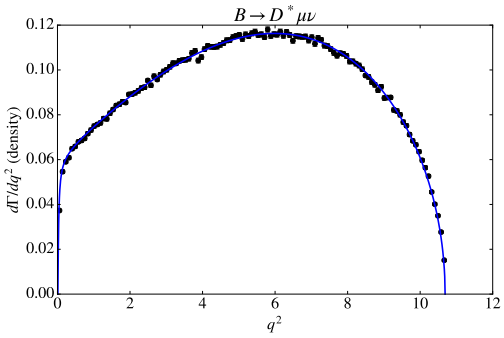
(b) $B \to D\mu\nu_\mu$

(c) $B \to D\tau\nu_\tau$

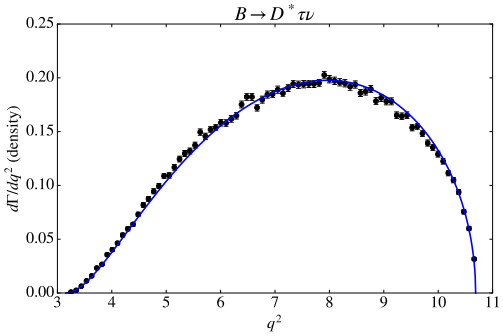
Figure 11.1: The default generated simulation data overlayed with the purported form factor model for $B \to D\ell\nu_\ell$.



(a) $B \to D^* e\nu_e$

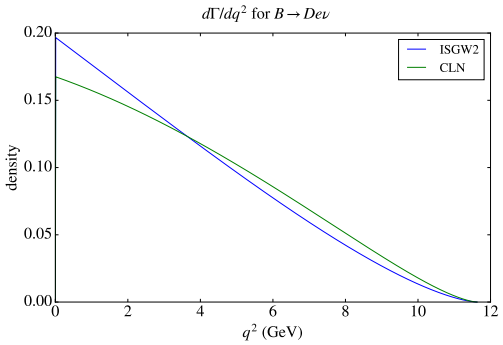


(b) $B \to D^* \mu\nu_\mu$

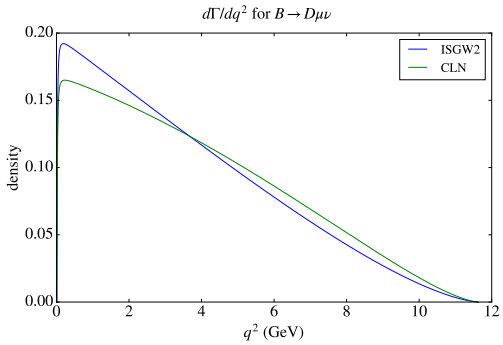


(c) $B \to D^* \tau\nu_\tau$

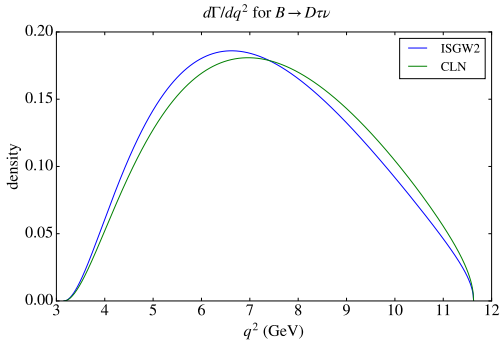
Figure 11.2: The default generated simulation data overlayed with the purported form factor model for $B \to D^* \ell\nu_\ell$.



(a) $B \to De\nu_e$

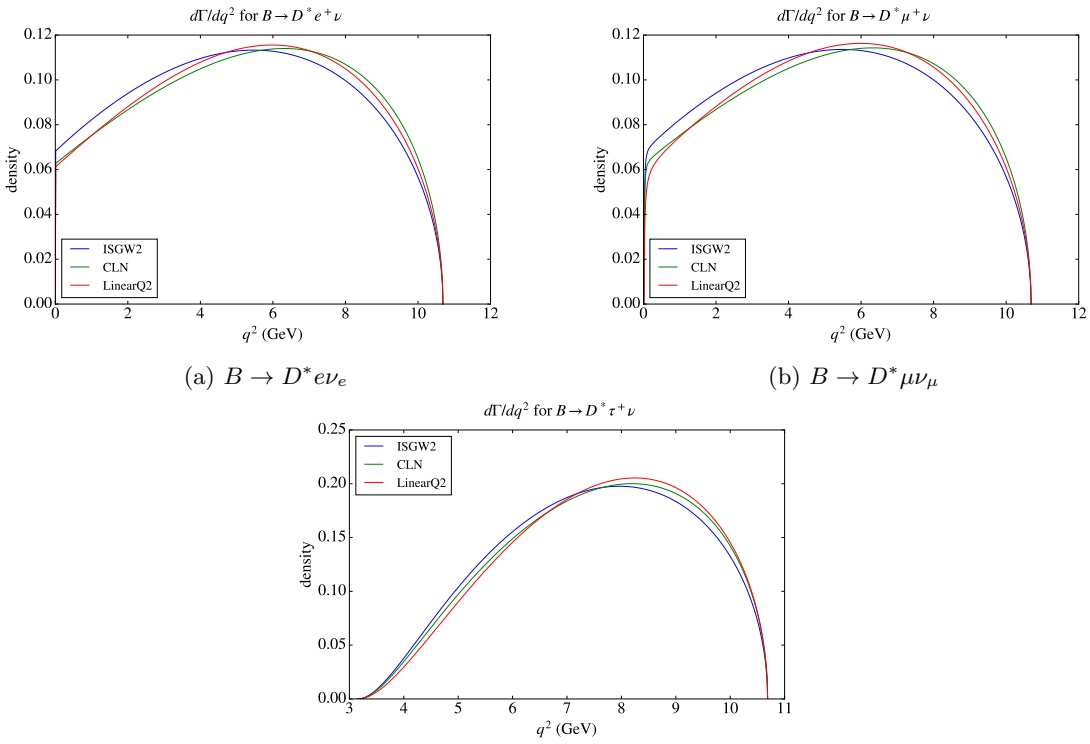


(b) $B \to D\mu\nu_\mu$



(c) $B \to D\tau\nu_\tau$

Figure 11.3: The differential decay rates for the $B \to D$ form factors considered in this analysis.

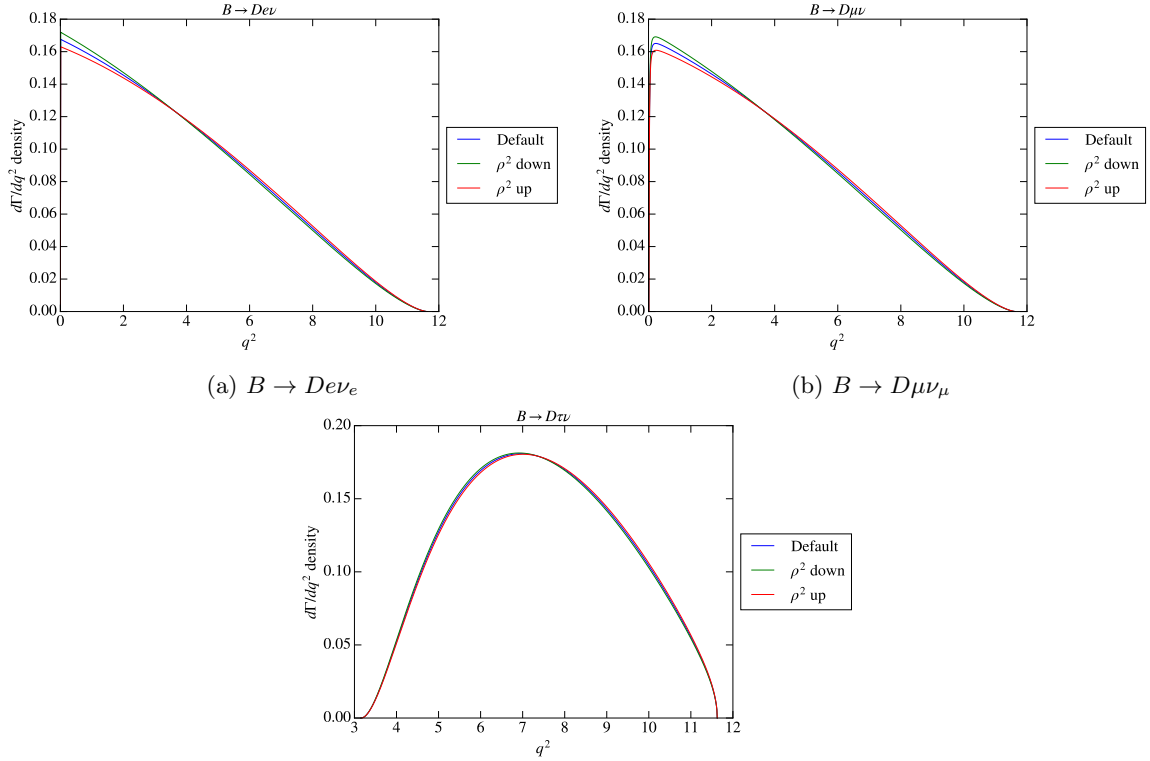


(a) $B \to D^* e \nu_e$

(b) $B \to D^* \mu \nu_\mu$

(c) $B \to D^* \tau \nu_\tau$

Figure 11.4: The differential decay rates for the $B \to D^*$ form factors considered in this analysis.

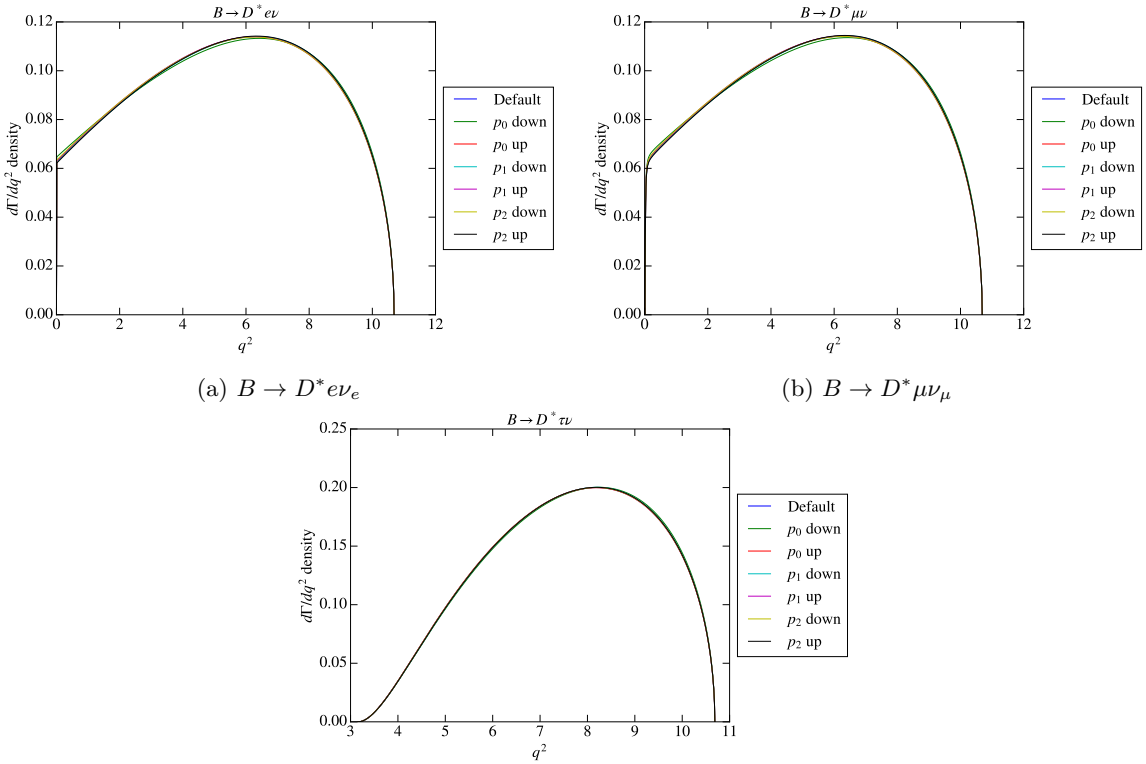


(a) $B \to De\nu_e$

(b) $B \to D\mu\nu_\mu$

(c) $B \to D\tau\nu_\tau$

Figure 11.5: The differential decay rates for the variations of $B \to D$ form factors considered in this analysis.



(a) $B \to D^* e\nu_e$

(b) $B \to D^* \mu\nu_\mu$

(c) $B \to D^* \tau\nu_\tau$

Figure 11.6: The differential decay rates for the variations of $B \to D^*$ form factors considered in this analysis. p_i, where $i = 0, 1, 2$, refer to the principal axes of the covariance matrix for (ρ^2, R_1, R_2) along which the variations are performed.

11.2.2 Uncertainties due to $B \to D^{**}\ell\nu_\ell$ form factors

We consider the impact of form factor models for semileptonic B decays to the lightest as well as the most abundant D^{**} mesons: D_0^*, D_1, D_1', and D_2^*. We consider re-weighting in the following kinematic variables:

- w: The four-velocity product between the B meson and the D^{**} meson. Like q^2, this sets the energy scale for which hadronization of the D^{**} meson takes place; dynamics at lower w are better understood than those at higher values.
- θ_ℓ: Angle of the three momentum of the ℓ in the virtual W's rest frame.

Obtaining the differential decay rates are, in principle, similar to those for $B \to D^{(*)}$. However, the models that we re-weight to are more complicated and have far more parameters. Therefore, we largely take the settings specified in BAD (1586). The following are the form factor model settings used in this analysis:

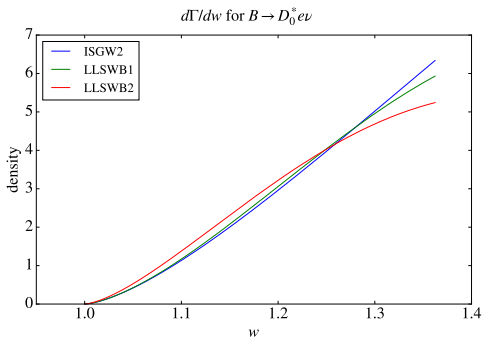
- Settings used to generate the simulated data:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: ISGW2[34].
- Settings re-weighted to in obtaining the central value:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: LLSW B1 [36].
- Settings re-weighted to in obtaining the systematic uncertainty:
  - $B \to D^{**}\ell\nu$, $\ell = e, \mu, \tau$: LLSW B2 [36].

Figures 11.7 show the different form factor models that are considered for $B \to D^{**}$ decays.

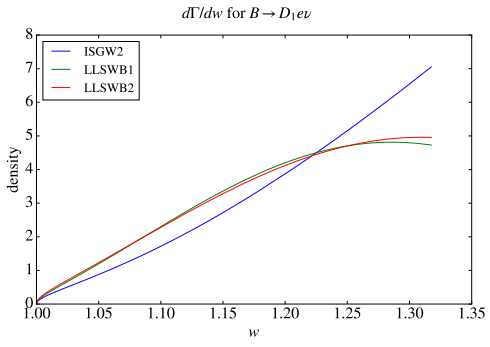
The method that we use to obtain the central value and systematic uncertainty is the same as those in $B \to D^{(*)}\ell\nu$ decays. The results are shown in table 11.2

Signal Type	Simulated data	Detector data
$D\tau$	0.0001	0.0056
$D^*\tau$	0.00004	0.0030

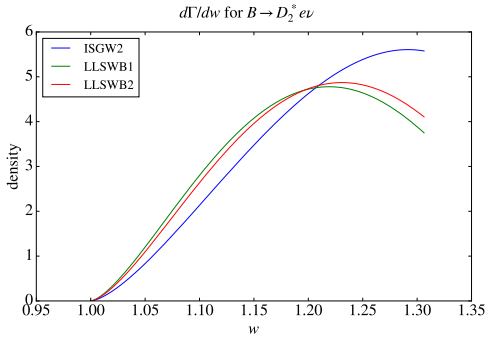
Table 11.2: Systematic uncertainties on $\hat{p}_j$ due to $B \to D^{**}$ form factors' influence on $\hat{f}_j$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.



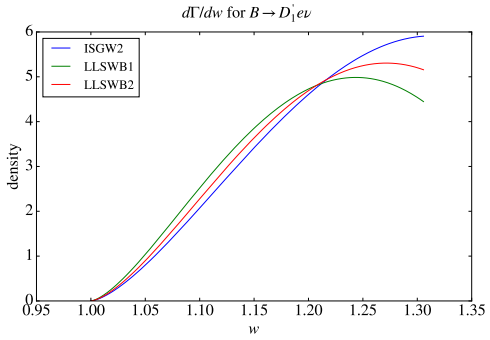
(a) $B \to D_0^* e\nu_e$



(b) $B \to D_1 e\nu_e$



(c) $B \to D_2^* e\nu_e$



(d) $B \to D_1' e\nu_e$

Figure 11.7: The differential decay rates for the $B \to D^{**}$ form factors considered in this analysis.

11.3 Branching fractions

The simulated data is generated based on a fixed set of branching fractions that are set to values consistent with the best knowledge at the time. Their settings influence our results in two ways:

1. Different decay modes generally have different distributions in Z_1 and Z_2. Changing their relative compositions can thus influence $\hat{f}_j$, which then affect the extraction of $\hat{p}_j$.

2. Different decay modes generally have different efficiencies. Changing their relative compositions can thus influence the overall event type efficiency $\hat{\epsilon}_j$.

There are two ways in which the default settings err. They are the following:

1. The default settings were set some number of years ago, and many of them inevitably become outdated. For instance, even the $B \to D^*(e|\mu)\nu$ branching fraction, which we have always thought to be well understood, was set to approximately 15% higher than what we find in the PDG today.

2. Even if the central values were set correctly, they are still, all known only to within some precision. Their potential variations from the nominal value must also be considered.

Of course, studying how every possible branching fraction can influence our results is neither practical nor necessary. Instead, we focus primarily on B decay modes[4] and consider only the following two types of decays:

1. B semileptonic D^{**} decays: among the resonant D^{**} decays, consider only decays to the 4 lightest D^{**} mesons, which are D_0^*, D_1, D_1', and D_2^*. Among non-resonant D^{**} decays, consider only those that satisfy the criteria in the next item. These cover the great majority of all B semileptonic D^{**} decays.

2. Among the most abundant B decays that pass to the final simulated data sample, consider only as many as is necessary to cover 50% of all records. Operationally, we rank the abundance of each mode in the simulated sample, and compute a cumulative proportion starting from the most abundant mode; all modes with a cumulative proportion less than 50% are considered.

Tables 11.3 and 11.4 list all the modes that we consider and their branching fractions from PDG, HFAG, and predictions[18][11][37], which we presume are considered as the most precise values by the community, along with the `DECAY.DEC` values. More specifically, we'd like to note to the readers that the branching fractions we correct to assume isospin symmetry, which result in lower uncertainties. We note that modes that are not well determined (i.e. uncertainty on the branching fraction is large) consist mostly of B semileptonic decays to resonant D^{**} mesons.

For the modes listed in tables 11.3 and 11.4, we update each record from those in `DECAY.DEC` to the present day PDG values by re-weighting. These weights are derived for each default simulated record as follows:

1. Traverse the simulated particle graph in search for modes listed in 11.3 and 11.4.

2. For each mode found, assign a weight $w = w_{PDG}/w_{\texttt{DECAY.DEC}}$.

3. The correction for the record is the product of all such computed weights.

To obtain the central value of our result, we use these re-weighted records to learn the estimated densities and to derive the signal efficiencies.

In addition, we correct for the asymmetry in production rate of B^+ vs. B^0 from decays of $\Upsilon(4S)$, based on the latest value from HFAG, by simply re-weighting the luminosity weight, which were initially calculated assuming equal production rate, as follows:

[4] We actually also assess the systematic uncertainty due to the D decay mode branching fractions, denoting it as a cross-check and will be discussed in later sections

Let ω_+ be the initial luminosity weight of a $B^\pm$ event, which is defined as $\omega_+ = \mathcal{L}_{\rm data}/\mathcal{L}_{\rm MC,B+}$, where $\mathcal{L}$ refers to luminosity. Since initially $\mathcal{L}_{\rm MC,B+} = 0.5 \times \mathcal{L}_{\rm MC,B}$, where B refers to both charged and neutral B's, and it clear that the corrected luminosity weight for charged and neutral B events should be:

$$\omega'_+ = \frac{\omega_+}{2 \times 0.513} \tag{11.6}$$

and

$$\omega'_0 = \frac{\omega_0}{2 \times 0.487}, \tag{11.7}$$

where the production rate of neutral B's used is 0.487 ± 0.006[5].

To quantify systematic uncertainties due to variations in the estimated densities, we proceed as follows:

1. Decide on a set of variations on the branching fractions (e.g. $D\ell\nu$ and $D^*\ell\nu$ decays are varied simultaneously based on their correlations).
2. For each record, re-compute the weights due to these variations and re-estimate the event type densities.
3. Use these densities to extract the result, and quote its difference from the central value as a systematic uncertainty.

To quantify systematic uncertainties due to variations in the signal efficiencies, we proceed as follows:

1. Decide on a set of variations on the branching fractions.
2. For each record, re-compute the weights due to these variations.
3. Compute the efficiency as usual, this time applying the new weights. Quote the difference between this and the central value as a systematic uncertainty.

The following sections will apply the procedures above for several variations of the branching fractions.

[5]It should be noted that there is indeed an uncertainty on the production rate, which provides another source of systematic uncertainty on the final result. Nevertheless, since the uncertainty is quite small, we assess the impact of varying the production rate but do not include in our final set of systematics. This is discussed further in Section 11.7

Decay Mode	DECAY.DEC Value	Corrected Value	Correction Uncertainty	Group
$B^+ \to \overline{D}^{*0}\mu^+\nu_\mu$	0.0617	0.0531	0.0012	A[6]
$B^+ \to \overline{D}^{*0}e^+\nu_e$	0.0617	0.0531	0.0012	A
$B^+ \to \overline{D}^{0}\mu^+\nu_\mu$	0.0224	0.0230	0.0010	A
$B^+ \to \overline{D}^{0}e^+\nu_e$	0.0224	0.0230	0.0010	A
$B^+ \to D_s^{*+}D^{*0}$	0.0278	0.0171	0.0024	B
$B^+ \to D_s^{+}D^{0}$	0.0129	0.009	0.0009	B
$B^+ \to D_s^{+}D^{*0}$	0.0124	0.0082	0.0017	B
$B^+ \to D^0\rho^+$	0.0134	0.0134	0.0018	C
$B^+ \to \overline{D}^{*0}a_1^+$	0.01597	0.019	0.005	D
$B^+ \to \overline{D}_1^0 e^+\nu_e$	0.0056	0.0096	0.001	E
$B^+ \to \overline{D}_0^{*0} e^+\nu_e$	0.0049	0.0044	0.0008	E
$B^+ \to \overline{D}_2^{*0} e^+\nu_e$	0.003	0.003	0.0004	E
$B^+ \to \overline{D}_1^{'0} e^+\nu_e$	0.009	0.002	0.0005	E
$B^+ \to D^{*-}\pi^+e^+\nu_e$	0.0006	0.006	0.0006	E
$B^+ \to D^{-}\pi^+e^+\nu_e$	0.0019	0.0042	0.0006	E
$B^+ \to \overline{D}_1^0 \mu^+\nu_\mu$	0.0056	0.0096	0.001	E
$B^+ \to \overline{D}_0^{*0} \mu^+\nu_\mu$	0.0049	0.0044	0.0008	E
$B^+ \to \overline{D}_2^{*0} \mu^+\nu_\mu$	0.003	0.003	0.0004	E
$B^+ \to \overline{D}_1^{'0} \mu^+\nu_\mu$	0.009	0.002	0.0005	E
$B^+ \to D^{*-}\pi^+\mu^+\nu_\mu$	0.0006	0.006	0.0006	E
$B^+ \to D^{-}\pi^+\mu^+\nu_\mu$	0.0019	0.0042	0.0005	E
$B^+ \to \overline{D}_1^0 \tau^+\nu_\tau$	0.0013	0.001	0.00014	E
$B^+ \to \overline{D}_0^{*0} \tau^+\nu_\tau$	0.0013	0.0004	0.00015	E
$B^+ \to \overline{D}_1^{'0} \tau^+\nu_\tau$	0.002	0.00012	0.00005	E
$B^+ \to \overline{D}_2^{*0} \tau^+\nu_\tau$	0.002	0.00021	0.00004	E

Table 11.3: Dominant B^+ decay modes having reasonably well determined values.

11.3.1 Uncertainties due to varying B branching fractions

The branching fraction variations we consider here are those listed in tables 11.3 and 11.4.

We group these modes into five classes. Group A are the semileptonic B decays varied with correlations taken into account (A_0 and A_1 are variations along the two principal axes), group B are B decays to strange flavored mesons, group C(D) consists of B decays to $D\rho(D^*a)$, and group E are B decays to $D^{**}\ell\nu$ and non-resonant $D\ell\nu\pi$. Since their values and uncertainties are reasonably well understood, we consider the set of variations consisting of increasing and decreasing, separately for each group, their central values by one standard deviation.

Uncertainties in $\hat{p}_j$ due to variations in $\hat{f}_j$

The resulting systematic uncertainties are listed in table 11.5.

Uncertainties in $\hat{\epsilon}_j$

The resulting systematic uncertainties are listed in table 11.6.

[6]Group A is varied along the principal axis based on correlations used for the world average. Thus two variations is performed for this group, rather than just one for other groups.

Decay Mode	DECAY.DEC Value	Corrected Value	Correction Uncertainty	Group
$B^0 \to D^{*-}\mu^+\nu_\mu$	0.057	0.0493	0.0011	A
$B^0 \to D^{*-}e^+\nu_e$	0.057	0.0493	0.0011	A
$B^0 \to D^-\mu^+\nu_\mu$	0.0207	0.0213	0.0010	A
$B^0 \to D^-e^+\nu_e$	0.0207	0.0213	0.0010	A
$B^0 \to D_s^{*+}D^{*-}$	0.024	0.0177	0.0014	B
$B^0 \to D_s^{+}D^{*-}$	0.0126	0.008	0.0011	B
$B^0 \to D^{*-}D^{*0}K^+$	0.01	0.0106	0.0009	B
$B^0 \to D_s^{*+}D^-$	0.009	0.0074	0.0016	B
$B^0 \to D_s^{+}D^-$	0.009	0.0072	0.0008	B
$B^0 \to D^-D^{*0}K^+$	0.0049	0.0035	0.0004	B
$B^0 \to D^{*-}D^{*+}K^0$	0.007	0.0081	0.0007	B
$B^0 \to D_{s1}^{+}D^-$	0.0098	0.0005	0.00014	B
$B^0 \to D^+\rho^-$	0.0077	0.0078	0.0013	C
$B^0 \to D^{*+}a_1^-$	0.012	0.013	0.0027	D
$B^0 \to D_2^{*-}e^+\nu_e$	0.0023	0.0028	0.0004	E
$B^0 \to D_1^{'-}e^+\nu_e$	0.0083	0.0019	0.00046	E
$B^0 \to D_0^{*-}e^+\nu_e$	0.0045	0.00408	0.00074	E
$B^0 \to D_1^{-}e^+\nu_e$	0.0052	0.0089	0.000911	E
$B^0 \to \overline{D}^{*0}\pi^-e^+\nu_e$	0.0007	0.0048	0.0008	E
$B^0 \to \overline{D}^{0}\pi^-e^+\nu_e$	0.002	0.0042	0.0006	E
$B^0 \to D_1^{-}\mu^+\nu_\mu$	0.0052	0.0089	0.000911	E
$B^0 \to D_0^{*-}\mu^+\nu_\mu$	0.0045	0.00408	0.00074	E
$B^0 \to D_1^{'-}\mu^+\nu_\mu$	0.0083	0.0019	0.00046	E
$B^0 \to D_2^{*-}\mu^+\nu_\mu$	0.0023	0.0028	0.0004	E
$B^0 \to \overline{D}^{*0}\pi^-\mu^+\nu_\mu$	0.0007	0.0048	0.0008	E
$B^0 \to \overline{D}^{0}\pi^-\mu^+\nu_\mu$	0.002	0.0042	0.0006	E
$B^0 \to D_1^{-}\tau^+\nu_\tau$	0.0013	0.0009	0.00013	E
$B^0 \to D_0^{*-}\tau^+\nu_\tau$	0.0013	0.0003	0.00014	E
$B^0 \to D_1^{'-}\tau^+\nu_\tau$	0.002	0.00017	0.00005	E
$B^0 \to D_2^{*-}\tau^+\nu_\tau$	0.002	0.00013	0.00004	E

Table 11.4: Dominant B^0 decay modes having reasonably well determined values.

Signal Type	BF variation type	Simulated data	Detector data
$D\tau$	$A_0 + 1\sigma$	-0.0003	0.0008
$D^*\tau$	$A_0 + 1\sigma$	0.0007	-0.0006
$D\tau$	$A_0 - 1\sigma$	0.0003	0.0004
$D^*\tau$	$A_0 - 1\sigma$	-0.0007	0.0
$D\tau$	$A_1 + 1\sigma$	0.0	-0.0023
$D^*\tau$	$A_1 + 1\sigma$	0.0001	0.0037
$D\tau$	$A_1 - 1\sigma$	-0.0002	0.0017
$D^*\tau$	$A_1 - 1\sigma$	-0.0002	-0.0017
$D\tau$	B $+1\sigma$	0.0001	0.0036
$D^*\tau$	B $+1\sigma$	0.0002	-0.0022
$D\tau$	B -1σ	-0.0002	0.0017
$D^*\tau$	B -1σ	-0.0002	-0.0017
$D\tau$	C $+1\sigma$	0.0	0.0015
$D^*\tau$	C $+1\sigma$	0.0006	-0.0017
$D\tau$	C -1σ	-0.0002	-0.0021
$D^*\tau$	C -1σ	-0.0003	0.0025
$D\tau$	D $+1\sigma$	0.0003	0.0020
$D^*\tau$	D $+1\sigma$	0.0009	-0.0009
$D\tau$	D -1σ	-0.0003	-0.0004
$D^*\tau$	D -1σ	-0.0009	0.0004
$D\tau$	E $+1\sigma$	-0.0001	0.0015
$D^*\tau$	E $+1\sigma$	0.0001	-0.0010
$D\tau$	E -1σ	0.0003	0.0013
$D^*\tau$	E -1σ	0.0	-0.0008

Table 11.5: Systematic uncertainties on $\hat{p}_j$ due to varying well determined B decay branching fractions. The first column is the event type, the second column indicates the kind of branching fraction variation, the third column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data. Note that groups A_0 and A_1 roughly correspond to $D\ell\nu$ and $D^*\ell\nu$ branching fractions, respectively.

11.3.2 Uncertainties due to difference between exclusive and inclusive branching fractions of $B \to X_c\ell\nu$

It is well-known [38] that there is a discrepancy of $\sim$ 1.5% between the inclusive branching fraction of semileptonic B decays and the sum of exclusive branching fractions. This motivated a previous *BABAR* analysis (BAD 2593) to "fill" the gap with B decays to $D^{**}\,(D\pi\pi)\,\ell\nu$ decays.

To assess the uncertainty due to the discrepancy, the $D^{**}\,(D\pi\pi)\,\ell\nu$ signal MC samples generated for the previous analysis are first re-weighted to make up 1.5% of the training set *prior* to any reconstruction and data filtering has been applied. Then the records go through the same pipeline as all the other records and incorporated into the final training set, which are again re-weighted to keep the total number of training data points the same. The samples used are summarized in Table 11.7, where we mix the three D^{**} decays modes in equal proportions.

The new dataset with these decays added are then used to assess the change in the result in $\hat{p}_j$ in the same procedure as the assessments of the B branching fraction systematic uncertainties: Table 11.8 shows the results.

Signal Type	BF variation type	Simulated data	Detector data
$D\tau$	$A_0 + 1\sigma$	0.00003	same as simulation
$D^*\tau$	$A_0 + 1\sigma$	0.00004	
$D\tau$	$A_0 - 1\sigma$	0.00003	
$D^*\tau$	$A_0 - 1\sigma$	0.00004	
$D\tau$	$A_1 + 1\sigma$	0.00004	
$D^*\tau$	$A_1 + 1\sigma$	0.00004	
$D\tau$	$A_1 - 1\sigma$	0.00004	
$D^*\tau$	$A_1 - 1\sigma$	0.00004	
$D\tau$	B $+1\sigma$	0.00002	
$D^*\tau$	B $+1\sigma$	0.00001	
$D\tau$	B -1σ	0.00002	
$D^*\tau$	B -1σ	0.00001	
$D\tau$	C $+1\sigma$	0.00001	
$D^*\tau$	C $+1\sigma$	0.00001	
$D\tau$	C -1σ	0.00001	
$D^*\tau$	C -1σ	0.00001	
$D\tau$	D $+1\sigma$	0.00003	
$D^*\tau$	D $+1\sigma$	0.00003	
$D\tau$	D -1σ	0.00003	
$D^*\tau$	D -1σ	0.00003	
$D\tau$	E $+1\sigma$	0.00012	
$D^*\tau$	E $+1\sigma$	0.00012	
$D\tau$	E -1σ	0.00012	
$D^*\tau$	E -1σ	0.00012	

Table 11.6: Systematic uncertainties in $\hat{\epsilon}_j$ due to varying poorly determined B decay branching fractions. The first column is the event type, the second column indicates the kind of branching fraction variation, the third column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

11.3.3 Uncertainties due to D branching fractions

The effect of the branching fractions of D decay modes are also studied. Rather than performing fluctuations of the branching fractions as we have done for the B decays, we perform a cross-check by making a correction on the most common decay of D to $K\pi\pi$.

More explicitly, we re-weight the events by, for each D decays to $K\pi\pi$, $w = w_{\text{PDG}}/w_{\texttt{DECAY.DEC}} = 0.992$. Since the branching fractions of D decays are well-measured, we deemed fluctuation by its measurement error unnecessary.

Table 11.9 shows the resulting systematic uncertainties on $\hat{p}_j$.

11.4 $B\overline{B}$ background validation

No matter how realistically we try to simulate the various event types, it is impossible to be certain of its accuracy without validating against a sample of detector data. However, the lack of such a sample is precisely the reason we appeal to simulation in the first place.

Despite this conundrum, it is perhaps not unreasonable to compare against a sample of detector data that is closely related to, and/or behaves similarly. Such a cross check will provide qualitative validation to the extent that we trust the similarities of such a "control sample" to an authentic sample.

In this analysis, all event type distributions other than those of continuum are taken from simulated data.

Decay Type	# event [10^6]	*BABAR* Dataset Name
$B^+ \to D_1(D\pi\pi)\ell\nu$	6.642	SP-11459-R24
$B^0 \to D_1(D\pi\pi)\ell\nu$	7.100	SP-11465-R24
$B^+ \to D_1(D^*\pi\pi)\ell\nu$	6.480	SP-11460-R24
$B^0 \to D_1(D^*\pi\pi)\ell\nu$	6.870	SP-11466-R24
$B^+ \to D(2S)(D\pi\pi)\ell\nu$	6.776	SP-11461-R24
$B^0 \to D(2S)(D\pi\pi)\ell\nu$	6.826	SP-11467-R24
$B^+ \to D(2S)(D^*\pi\pi)\ell\nu$	6.530	SP-11462-R24
$B^0 \to D(2S)(D^*\pi\pi)\ell\nu$	6.769	SP-11468-R24
$B^+ \to D(2S)^*(D\pi\pi)\ell\nu$	6.369	SP-11463-R24
$B^0 \to D(2S)^*(D\pi\pi)\ell\nu$	6.552	SP-11469-R24
$B^+ \to D(2S)^*(D^*\pi\pi)\ell\nu$	6.425	SP-11464-R24
$B^0 \to D(2S)^*(D^*\pi\pi)\ell\nu$	6.616	SP-11470-R24

Table 11.7: Signal MC samples used for assessing the gap between inclusive and sum of exclusive $B \to X_c\ell\nu$ braching fractions.

Signal Type	Simulated data	Detector data
$D\tau$	0.0001	0.0006
$D^*\tau$	0.0003	0.0007

Table 11.8: Systematic uncertainties in $\hat{p}_j$ due to the discrepancy between inclusive and the sum of exclusive branching fractions of B decays to $X_c\ell\nu$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

Although we concede the absence of a control sample for $D\tau$ and $D^*\tau$ event types, we propose using records that satisfies at least one of the following conditions to be the "sideband control sample" for the $B\overline{B}$ event types:

S1. $E_{extra} \geq 3$ GeV.

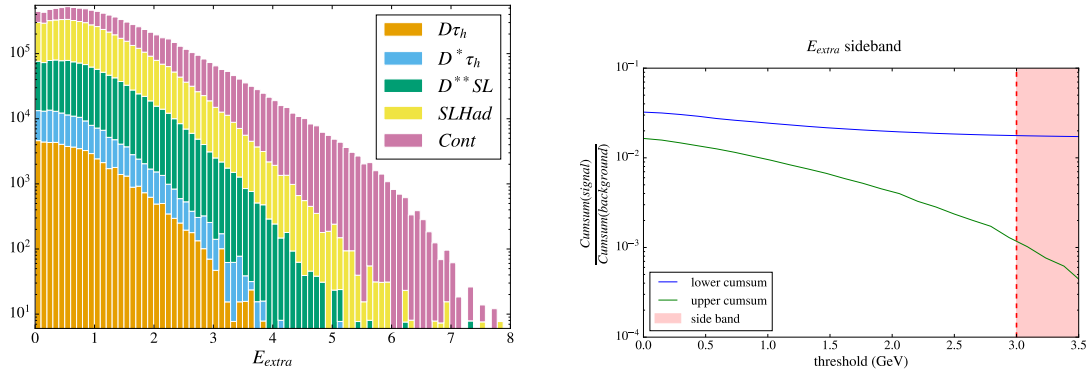
S2. $|\vec{p}_h^{sig}| \geq 2.3$ GeV.

Figures 11.8, and 11.9 show that this sample consists almost entirely of event types $B\overline{B}$ and continuum. Furthermore, compared to those records not in the sideband sample, the proportion of the signal types contribution is at least an order of magnitude less.

11.4.1 Qualitative validation

Figure 11.10 shows the Z_1 and Z_2 comparisons for data points belonging to the sideband. Overall, the distribution shape for the on peak data points (black) and the stacked histograms agree reasonably well. One could argue, however, that the residuals of the Z_2 score is possibly showing signs of a slight upward trend. The normalization is slightly different, but this not as much a concern since they are afloat in the optimization.

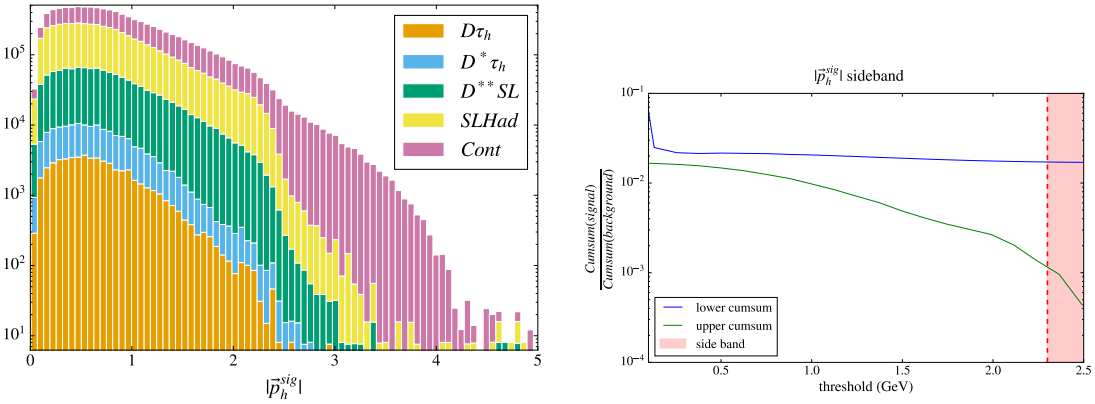
We show the sideband comparisons for the raw features used to train Z_1 and Z_2 in appendix C; they generally show good distribution shape agreement as well.



(a) Stacked log counts.

(b) Cumulative sum ratios.

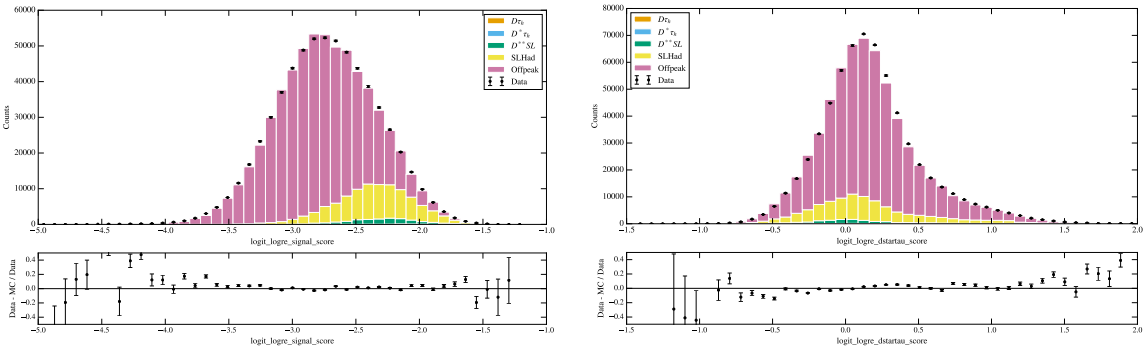
Figure 11.8: Sideband sample in E_{extra}.



(a) Stacked log counts.

(b) Cumulative sum ratios.

Figure 11.9: Sideband sample in $|\vec{p}_h^{sig}|$.



(a) Sideband signal score (Z_1).

(b) Sideband $D^*\tau$ score (Z_2).

Figure 11.10: Sideband comparison for Z_1 and Z_2. All data points used to produce these figures belong to the sideband sample. The black points are on-peak detector data, the pink filled histograms are luminosity scaled off-peak data, and the yellow and green filled histograms are the simulated $B\overline{B}$ background.

Signal Type	Simulated data	Detector data
$D\tau$	0.0	0.0008
$D^*\tau$	0.0001	0.0001

Table 11.9: Systematic uncertainties in $\hat{p}_j$ due to corrections to branching fraction of $D \to K\pi\pi$. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

11.4.2 Quantitative validation and its uncertainty

We can attempt to quantify, by using the sideband samples, a systematic uncertainty on $\hat{p}_j$ due to possible simulation misrepresentations of the $\hat{f}_j$'s in $B\overline{B}$ background event types by making the following assumptions:

A1. The off-peak data is a good representation of the continuum event type. "Good" in this case roughly means that any discrepancies is sub-dominant compared to discrepancies seen in the sideband samples.

A2. The signal event types are negligible in the sideband. Together with the point above, all discrepancies in the sideband are assumed to be due to the $B\overline{B}$ background.

A3. The discrepancy in Z_1 and Z_2 is the same independent of whether a record is in the sideband or not.

We define a weight function $w(z_1, z_2)$ that rectifies the discrepancy of the $B\overline{B}$ component in the sideband sample as follows:

1. Let $g(z_1, z_2)$ be the density function in Z_1 and Z_2 for the on-peak sideband. By assumption **A2.**, we can decompose this into two components: contributions from continuum and contributions from $B\overline{B}$ background. We write this as
$$g(z_1, z_2) = (1 - p_{cont})g_{B\overline{B}}(z_1, z_2) + p_{cont}g_{cont}(z_1, z_2) \tag{11.8}$$
2. Let $f_{B\overline{B}}(z_1, z_2)$ be the density function in Z_1 and Z_2 for the simulated $B\overline{B}$ background that is also in the sideband.
3. Define the weight function as follows:
$$w(z_1, z_2) = \frac{g(z_1, z_2) - p_{cont}g_{cont}(z_1, z_2)}{(1 - p_{cont})f_{B\overline{B}}(z_1, z_2)} \tag{11.9}$$

Finally, we form the estimate $\hat{w}(z_1, z_2)$ by estimating each quantity in equation 11.9 as follows:

- Estimate $g(z_1, z_2)$ using a kernel density estimate trained on the on-peak sideband sample.
- By assumption **A1.**, we estimate $g_{cont}(z_1, z_2)$ using a kernel density estimate trained on the off-peak sideband sample.
- Estimate $f_{B\overline{B}}(z_1, z_2)$ using the simulated sideband $B\overline{B}$ background.
- There is no firm basis from which to estimate p_{cont}. We can, in principle, derive a family of weight functions that are parameterized by a range of reasonable settings for p_{cont}. However, for simplicity, we set p_{cont} to those derived in the simulation as an initial guess.

By assumption **A3.**, we use $\hat{w}(z_1, z_2)$ to quantify a systematic uncertainty as follows:

1. Apply a weight $w(z_1, z_2)$ for each simulated $B\overline{B}$ background data point.
2. Use these weighted points to update the estimated density functions input into the signal extraction.
3. Quote the difference between the updated optimized value and the central value as the systematic uncertainty.

11.4.3 Results

Signal Type	Simulated data	Detector data
$D\tau$	0.0024	0.0017
$D^*\tau$	0.0001	0.0003

Table 11.10: Systematic uncertainties on $\hat{p}_j$ due possible misrepresentations of $\hat{f}_j$ by the simulated $B\overline{B}$ background. The first column is the event type, the second column is the systematic derived from simulated data, while the last column is the systematic derived from the detector data.

Table 11.10 show the systematic uncertainties quantified using the procedure discussed above. While this approach may seem plausible, we recommend taking its results with a grain of salt. The most brittle assumption, in our opinion, is **A3.**; this alone is enough to make us feel more comfortable had we only proceeded with the qualitative comparisons. Even more than that, we have avoided discussing the subtle numerical issues involved with estimating $w(z_1, z_2)$ when the densities are nearly zero. The origin of this difficulty is statistical: sideband statistics are limited, especially in regions rich in signal. It is, therefore, not surprising that such estimations are approximate at best.

11.5 Detector efficiencies

Another factor that influences efficiencies is the detector response. Since we estimate $\hat{\epsilon}_j$'s entirely from simulation, we must quantify the uncertainties due to its misrepresentation of the real detector.

It is not difficult to imagine the numerous places that a simulated detector could misrepresent a real detector. However, knowing where all such errors could occur is, once again, neither practical or necessary. Like many analyses before us, we focus only on the net effect of these misrepresentations. The two aspects that we believe summarizes the influence of such effects come in the form of an uncertainty on the reconstruction efficiency. They are the following:

1. Tracking efficiency: the probability that a track is reconstructed is different in a real detector. The uncertainty on the reconstruction efficiency of a physics event is, therefore, related to the cumulative uncertainty on the reconstruction efficiency of its tracks.

2. Particle identification (PID) efficiency: the probability that a given particle is identified as a certain species is also different in a real detector. Since reconstruction relies on the presence of identified particles, the cumulative efficiency of successfully identifying each required particle affects the reconstruction efficiency of the given physics event.

The following subsections will describe how we obtain estimates for both types of efficiencies listed above. We will discover that these uncertainties are far smaller than those discovered in section 11.3, and so we do not pursue more detailed studies at this time.

11.5.1 Tracking efficiency uncertainty

The tracking group recommends that we assign an uncertainty of 0.2% for every track that is present in a physics event. This convenience is possible due to the significant improvements in the tracking algorithms used since release 24 of the *BABAR* framework software. Prior to these improvements, analysts would have to lookup a set of tables to obtain the correct uncertainties for every track in an event.

For event type j, we assign the systematic uncertainty on $\hat{\epsilon}_j$ due to tracking as follows:

$$\sigma^{trk}_{\hat{\epsilon}_j} = \frac{\sum_{i=1}^{N^{(j)}} w_i^{(j)} n_i^{(j)} 0.2\%}{\sum_{i=1}^{N^{(j)}} w_i^{(j)}} \tag{11.10}$$

where we have defined the following:

- $N^{(j)}$: number of records in the data sample belonging to event type j.
- $w_i^{(j)}$: the weight of record i for records in the data sample belonging to event type j.
- $n_i^{(j)}$: the number of tracks of record i for records in the data sample belonging to event type j.

In other words, we simply take the weighted average of the recommended total tracking uncertainty over records of a specified event type. It turns out that the uncertainties for both signal event types are $\boxed{2.3\%}$.

11.5.2 PID efficiency uncertainty

The PID group recommends applying a per particle uncertainty by looking up a set of PID lookup tables for each identified particle that is also used in the reconstruction of a record. They also suggest, in lieu of the lookup table approach, the approximate uncertainties listed in table 11.11.

Particle	Uncertainty
e	0.7%
μ	1.3%
K	1.1%

Table 11.11: Per particle PID efficiency uncertainty for each particle species.

A typical reconstruction of our records consists of the following identified particles:

- Two leptons. One for the semileptonic reconstruction of the B_{tag} and another for the leptonic veto of the reconstructed τ daughter. For an average case estimate, we simply take 1 electron and 1 muon for a total uncertainty of 2.0%.
- Two kaons, one for each D meson reconstruction. This gives a total uncertainty of 2.2%.

The estimates above totals to a $\boxed{4.4\%}$ uncertainty on the PID efficiency uncertainty for each event type.

11.6 Systematic Uncertainty on the Bias Correction

The bias correction procedure as outlined in Section 9.4 has an associated systematic uncertainty stemming from our limited knowledge of the background proportions (*i.e.* p_i where $i = D^{**}SL(3), Comb(4), Cont(5)$).

As stated before, the `mc.central` simulated data represents our best educated guess of the detector data. The bias correction attempts to rectify the effect of the difference in signal proportions, and in this section we aim to quantify the effects of differing background proportions by making the following assumptions:

1. The uncertainty in the background proportions are dominated by branching fraction measurement uncertainty.
2. Uncertainty in the proportion of continuum component is negligible since we use the off-peak data, which can be assumed to be identical to the continuum component of the real data.

The systematic uncertainty due to the branching fractions can be quantified as before by fluctuating the background proportion of $D^{**}SL$ and $Comb$, and measuring the bias as before. The changes in the bias from these flucations will used to upper bound the systematic uncertainty. The next few sections aims to quantify how much we trust the values generated by the `mc.central` model.

McBType	w	Fraction
Dstarstar_res	7062.35	0.34
Had	6879.70	0.33
Dstarstar_nonres	3799.62	0.18
Dstarl	1895.73	0.09
Dl	836.97	0.04
Dstartau	116.14	0.01
Dtau	46.43	0.00
SL	30.95	0.00

(a) Event type 3, $D^{**}SL$.

McBType	w	Fraction
Had	10164.59	0.57
Dstarl	4794.70	0.27
Dl	2297.09	0.13
Dstartau	223.51	0.01
Dtau	137.01	0.01
SL	126.56	0.01

(b) Event type 4, $Comb$.

Figure 11.11: Ranking of the `McBTypes` in 10000 records of event types 3 and 4, or $D^{**}SL$ and $Comb$.

11.6.1 Branching Fraction Uncertainty in Background Components

To quantify the total uncertainty in p_3 and p_4 due to branching fraction measurement errors, we rank the most common decay modes for each event type. More specifically, we draw a random sample of 10000 records for each event type, then categorize each B into one of 9 possible `McBTypes`, or categories:

- `NoB`: Not a B decay.
- `Dtau`: $B \to D\tau\nu$.
- `Dstartau`: $B \to D^*\tau\nu$.
- `Dl`: $B \to D\ell\nu$.
- `Dstarl`: $B \to D^*\ell\nu$.
- `Dstarstar_res`: $B \to D^{**}\ell\nu$.
- `Dstarstar_nonres`: $B \to D^{**}\ell n\pi\nu$.
- `SL`: All other semileptonic decays of B.
- `Had`: Hadronic decays of B.

Figure 11.11 shows the breakdown of the background event types, and Table 11.12 shows the upper bounds of relative measurement errors for each `McBType` based on Tables 11.3 and 11.4.

McBType	Relative Measurement Error
Dl	1%
Dstarl	1%
Dstarstar_res [7]	15%
Dstarstar_nonres	10%
SL	2%
Had	10%

Table 11.12: Summary of relative uncertainties on the most common decay modes in the background event types.

[7] It should be noted that other anaylses, for instance [2], assign much more conservative uncertainties on the branching fractions of B into D^{**} decay modes, up to 100%.

From these values, we can proceed to bound the relative uncertainty on p_3 and p_4 by considering them as a weighted sum of the 9 `McBTypes`. This results in 6.3% and 5.7% uncertainty on $p_{D^{**}SL}$ and p_{Comb}, respectively.

The following recipe is used to assign systematic uncertainties due to the choice of background proportions:

1. Calculate the bias for a grid of points in true ($D\tau$, $D^*\tau$) space.
2. For each grid point (i.e. choice of true signal proportions), we vary the background proportions up and down based on the values given above.
3. Assign the systematic uncertainty as the maximum difference between the biases calculated from the four variations and the default bias.

When we performed the above procedure for 15 grid points ranging from $(0, 0)$ to $(0.02, 0.01)$ in step sizes of 0.0005, we found that the systematic uncertainty is rather uniform as a function of the true proportions. Thus, we assign as the systematic uncertainty the average of those 15 measurements.

Signal Type	Systematic Uncertainty
$D\tau$	0.0008
$D^*\tau$	0.0005

Table 11.13: Systematic uncertainties on the bias correction. The first column is the event type, the second column is the systematic uncertainty. There is no differentiating between simulated and detector data since all estimations are performed in simulation.

11.7 Additional sources of systematic uncertainty

11.7.1 Charged vs. neutral B production rate at $\Upsilon(4S)$

As discussed in a footnote previous, there is an uncertainty on the B^0 production rate of 0.006. We assess the systematic uncertainty due to raising and lowering of the production rate (i.e. 0.493 and 0.481) following the same steps as assessing systematic uncertainty due to branching fractions. This results in $\boxed{0.0001(0.0017)}$ and $\boxed{0.0004(0.0024)}$ uncertainty on $\hat{p}_{D\tau}$ and $\hat{p}_{D^*\tau}$, respectively for simulated(detector) test dataset.

11.7.2 Possible discrepancy of $R(D^{**})$

The main motivation behind this analysis is the tension between measured and predicted value of $R(D^{(*)})$ of around 30%. We assess a possible systematic uncertainty arising from discrepancy in the $R(D^{**})$ system of a similar magnitude.

We increase the branching fractions of semitauonic decays of B to D^{**} (i.e. the last 4 decays listed in Tables 11.3 and 11.4) by 30%. This results in $\boxed{\text{-}0.0002(\text{-}0.0003)}$ and $\boxed{\text{-}0.0004(0.0007)}$ uncertainty on $\hat{p}_{D\tau}$ and $\hat{p}_{D^*\tau}$, respectively for simulated(detector) test dataset.

11.8 Summary

Table 11.14 collects all the systematic uncertainties derived in this chapter.

		Simulation		Detector Data	
		$D\tau$	$D^*\tau$	$D\tau$	$D^*\tau$
$\hat{p}$ uncertainties	$D^{(*)}\tau$ form factors	0.0005	0.0014	0.0030	0.0024
	$D^{**}\tau$ form factors	0.0001	0.00004	0.0056	0.0030
	Semileptonic D^{**} branching fractions	0.0003	0.00005	0.0015	0.0010
	Semileptonic $D^{(*)}$ branching fractions	0.0004	0.0007	0.0031	0.0043
	Strange decay branching fractions	0.0002	0.0002	0.0036	0.0022
	$B \to D^{(*)}\rho$ branching fractions	0.0002	0.0006	0.0021	0.0025
	$B \to D^{(*)}a_1$ branching fractions	0.0003	0.0009	0.0020	0.0009
	$B\overline{B}$ background shape validation	0.0024	0.0001	0.0017	0.0003
	D decay branching fraction cross-check	0.00004	0.0001	0.0008	0.0001
	Gap sample for $B \to X_c\ell\nu$ branching fractions	0.0001	0.0003	0.0006	0.0007
	Background proportions on bias correction	0.0008	0.0005	same as simulation	
$\hat{\epsilon}$ uncertainties	Semileptonic D^{**} branching fractions	0.00012	0.00012		
	Semileptonic $D^{(*)}$ branching fractions	0.00005	0.00005		
	Strange decay branching fractions	0.00002	0.00001		
	$B \to D^{(*)}\rho$ branching fractions	0.00001	0.00001	same as simulation	
	$B \to D^{(*)}a_1$ branching fractions	0.00003	0.00003		
	Tracking efficiency	0.0003	0.0002		
	PID efficiency	0.0006	0.0005		

Table 11.14: Summary of systematic uncertainties on $\hat{p}_j$ and $\hat{\epsilon}_j$.

An immediate observation is that the systematic uncertainties computed using the detector data are much larger than those computed using the simulated data. In fact, the error size of the detector data is consistently larger by around a factor of 5. We will postpone a discussion regarding this difference to section 12.2. It is also interesting to note that the largest sources of systematic uncertainties are not due to D^{**}'s but due to the uncertainty in the branching fractions of non-D^{**}'s.

We suspect the reason that we are so susceptible to systematic variations is simply due to having very little signal. Small signals do not just pose challenges in distinguishing it from the stochasticity of the background, but also easily drowned out by systematic variations. One way to improve this analysis is to try to combine signal categories so that it is more robust against systematic variations in the background; for example, measuring the sum of $D\tau$ and $D^*\tau$ branching fractions. The challenge in this suggestion is to derive a quantity from the sum that still says something meaningful about $\mathcal{R}(D^{(*)})$.

Chapter 12

Results and Discussions

This chapter presents the final result. We will first present the unblinded result in the form of a confidence region, and conclude by some followup discussions.

12.1 Unblinded Results

The value of $\mathcal{R}(D^{(*)})$ can be calculated by plugging in the values of $\hat{p}_{D^{(*)}\tau}$ and $\hat{\epsilon}_{D^{(*)}\tau}$ into equations 4.10 and 4.15. The value of $\hat{\epsilon}_{D^{(*)}\tau}$ is computed using the prescription presented in 4.7; we find that they are

$$\epsilon_{D\tau} = 0.0129 \pm 0.0007(\text{sys.}) \tag{12.1}$$
$$\epsilon_{D^*\tau} = 0.0106 \pm 0.0005(\text{sys.}). \tag{12.2}$$

What remains is to compute $\hat{p}_{D^{(*)}\tau}$ by applying the procedure outlined in chapter 9 on the detector data and using the combined statistical and systematic uncertainties to construct a confidence region on $\mathcal{R}(D^{(*)})$. The prescription is summarized in these steps:

S1 Solve the optimization problem of equation 9.1, but this time plugging in the detector data into the objective.

S2 Now that $\hat{p}_{D^{(*)}}$ are computed, lookup the bias correction required in the table constructed in section 9.4. We can then compute the central value of $\mathcal{R}(D^{(*)})$ using the bias corrected $\hat{p}_{D^{(*)}}$ and $\hat{\epsilon}_{D^{(*)}\tau}$.

S3 Construct the confidence interval/region using the combined statistical and systematic uncertainty. We will outline how this is done below.

12.1.1 Results for $\hat{p}_{D^{(*)}\tau}$

We begin by applying step **S1**. The signal proportions obtained directly from the solution of equation 9.1, i.e. *before* bias correction, are

$$\hat{p}_{D\tau} = 0.0051 \pm 0.0005(\text{stat.}) \pm 0.0085(\text{sys.})$$
$$\hat{p}_{D^*\tau} = 0.0009 \pm 0.0001(\text{stat.}) \pm 0.0066(\text{sys.}) \tag{12.3}$$
$$\rho_{\hat{p}_{D\tau},\hat{p}_{D^*\tau}} = -0.21 \tag{12.4}$$

We now turn to applying step **S2**; figure 12.1 shows the bias table overlayed with the result that was obtained in step **S1**. Since the bias table covers the range of physical signal proportions, it is clear that we have obtained an unphysical value. This is actually not a problem in frequentist statistics, but we do give

up quoting a central value for $\mathcal{R}(D^{(*)})$[1]. This does not preclude us from constructing a confidence region, so we turn now to applying step **S3**.

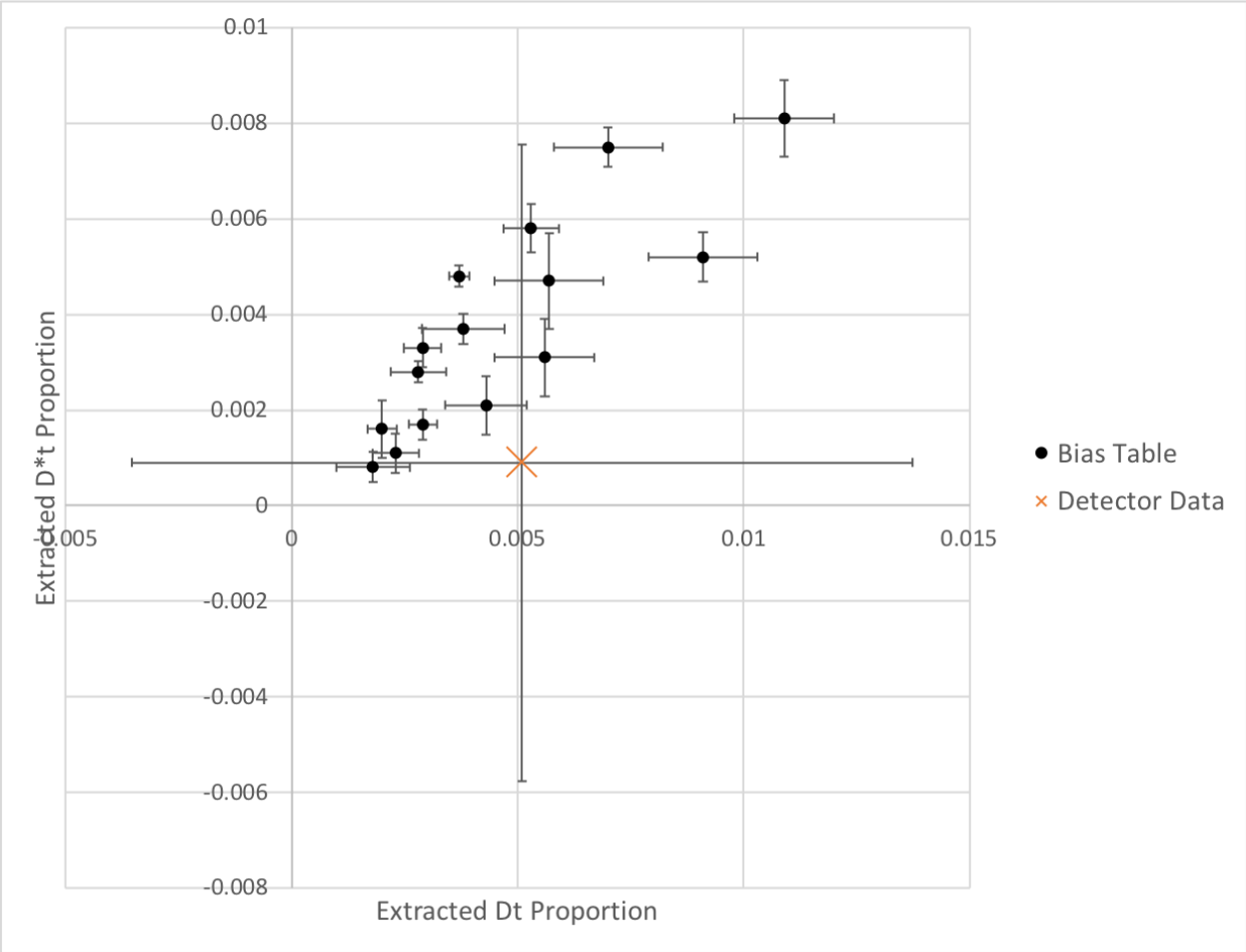


Figure 12.1: Black dots are $\hat{p}_{\text{sig}}$ of MC datasets with varying signal proportions with uncertainties as described in Sec. 11.6, and represents the bias table. Orange cross is the $\hat{p}_{\text{sig}}$ of the detector data with statistical and systematic uncertainties.

12.1.2 Confidence Region

To construct a confidence region over $\mathcal{R}(D^{(*)})$, first construct a confidence region over each grid point of the bias table. Once the confidence region is constructed for $\hat{p}$, simply apply the bias correction at each grid point and apply equations 4.10 and 4.15.

We apply the following procedure to obtain the confidence region over each grid point of the bias table. Denote the uncorrected fit result of step **S1** to be $\hat{p}$. For the ith grid point p_i,

1. Form the chi-squared statistic
$$\chi_i^2 = (p_i - \hat{p})^T M^{-1} (p_i - \hat{p}) \tag{12.5}$$
where M is the covariance matrix between p_i and $\hat{p}$.

[1]One could attempt to extrapolate a central value, but we thought this to be unreliable given the overall shape of the bias table.

2. Calculate the p-value as

$$p\text{-value}_i = 1 - F(\chi^2_i; 2), \tag{12.6}$$

where $F(x; k)$ is the CDF of the chi-squared distribution with k degrees of freedom.

Once all the p-values are computed, we simply mark a grid point as belonging to the 68% (95%) confidence region if its p-value exceeds 0.32 (0.05). Figure 12.2 shows p-value$_i$ for all points of the bias table while figure 12.3 shows the resulting confidence regions of $p_{D^{(*)}\tau}$. Finally, we transform the confidence regions in $p_{D^{(*)}\tau}$ to $\mathcal{R}(D^{(*)})$'s. The result is shown in figure 12.4, and we overlay the results from other measurements in figure 12.5. It is clear that our result is not sufficiently sensitive to provide any evidence for or against the existing discrepancies.

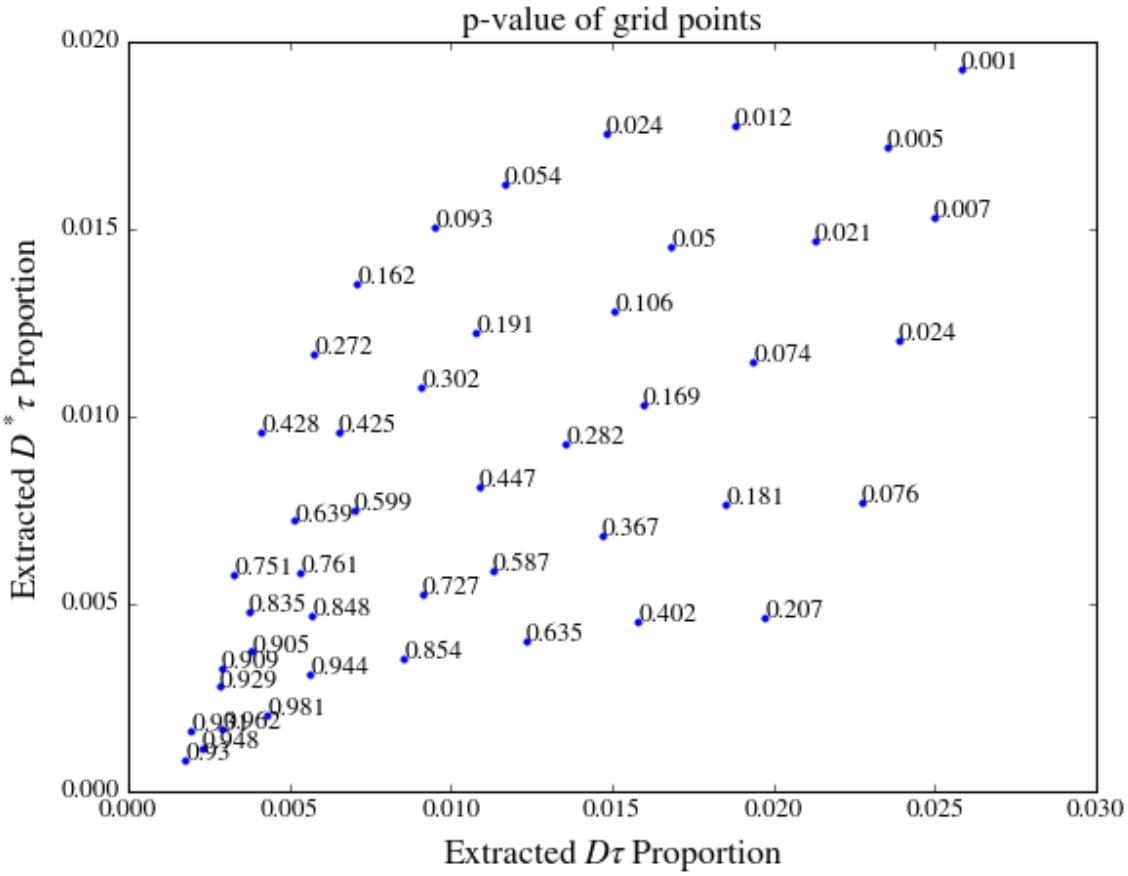


Figure 12.2: Points of the bias tables with their p-values as labels.

12.2 Discussions

We now discuss the sources of the large uncertainties that are observed and suggest some improvements. It is clear that we are, by far, systemtics dominated. This section will therefore focus the discussion on the large systematic uncertainties.

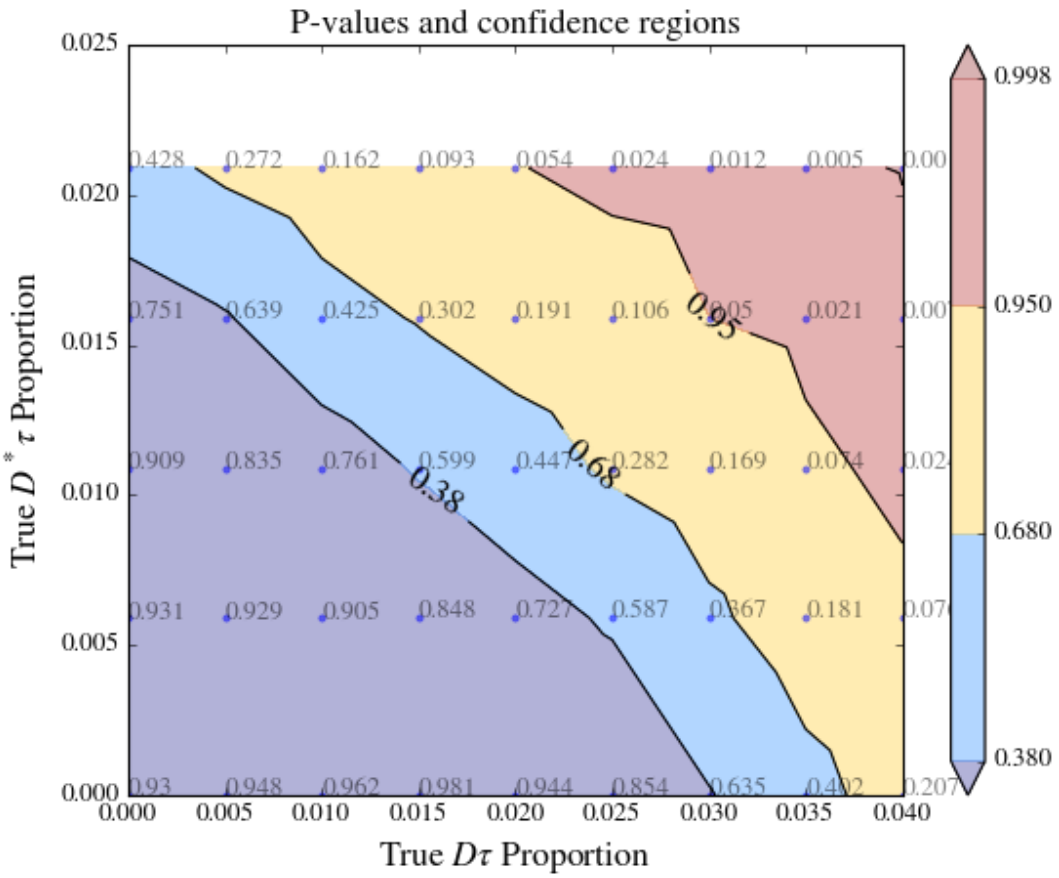


Figure 12.3: Confidence regions of $p_{D^{(*)}\tau}$

The most glaring problem is that the systematic uncertainties are far larger when applied to data than when they are applied to simulation. In fact, the detector data column in table 11.14 is consistently 5x larger than those in the simulated data. Such a large asymmetry between the two columns suggest that the simulation could be a rather poor representation of the data. If this were the case, then an analysis such as ours that optimizes aggressively on simulation might in fact adversely impact performance in real data. It is of course unrealistic for an analysis to take no inputs from simulation, but one could be more active about avoiding the aspects of simulation that are likely to be poorly simulated.

Another observation is that our signal is a lot smaller than the background compared to similar analyses [1][2][3][4], which suggests that we are potentially more sensitive to systematic variations in the background. We were optimisitic early on in our analysis because we have achieved great suppression in statistical errors; but in the end, the systematics proved to be the more difficult to control and cannot be tamed purely by statistical techniques.

One can also argue that we might not have done enough to improve the signal to background ratio. While it is true that we have already done a reasonable job in separating the signal from background by using the classifiers, it is quite possible that we could've benefit from courser cuts further upstream in the analysis that might have given us a cleaner sample to work with; e.g. cleaning the construction of the tag side B more carefully.

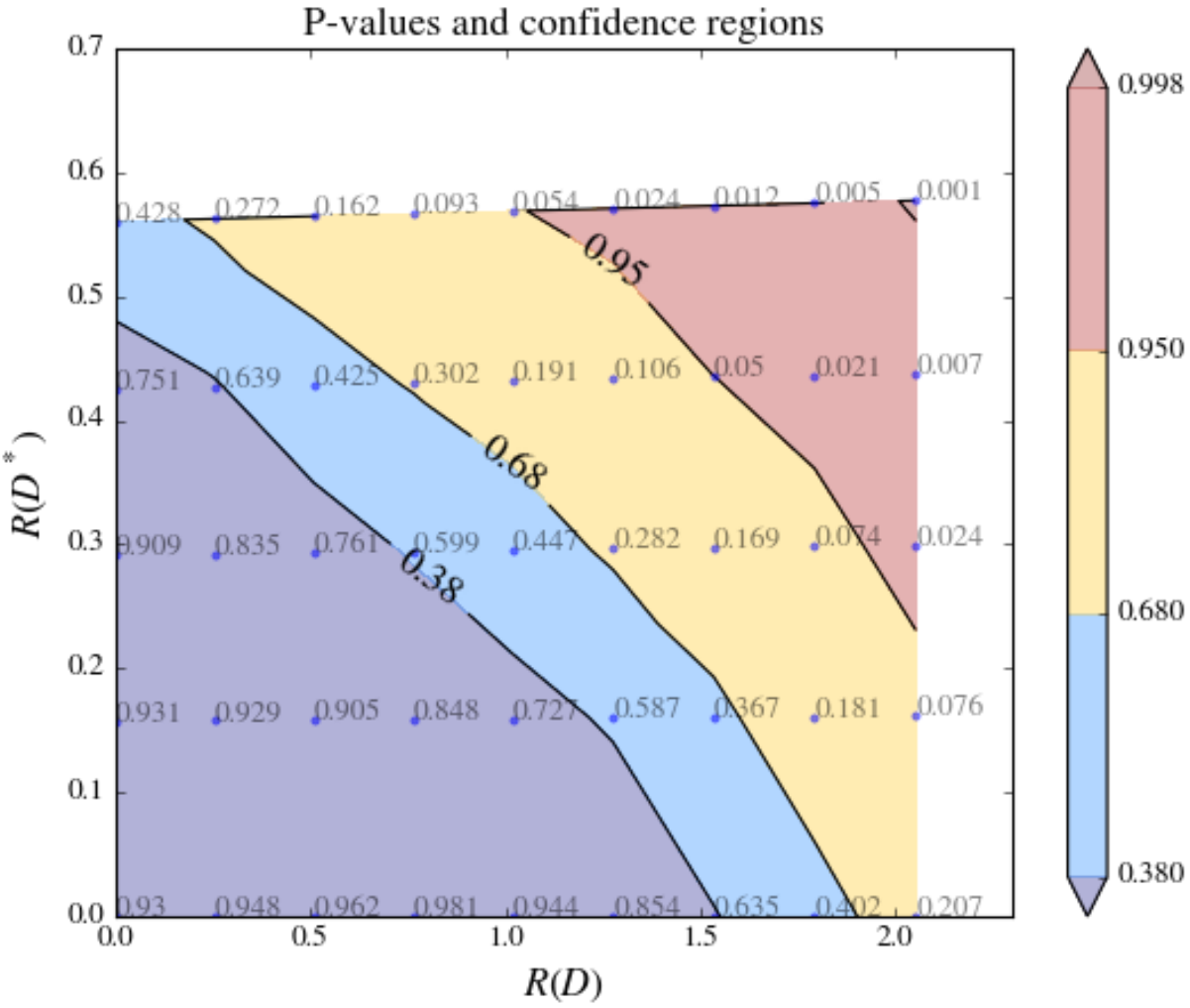


Figure 12.4: Confidence regions of $\mathcal{R}(D^{(*)})$

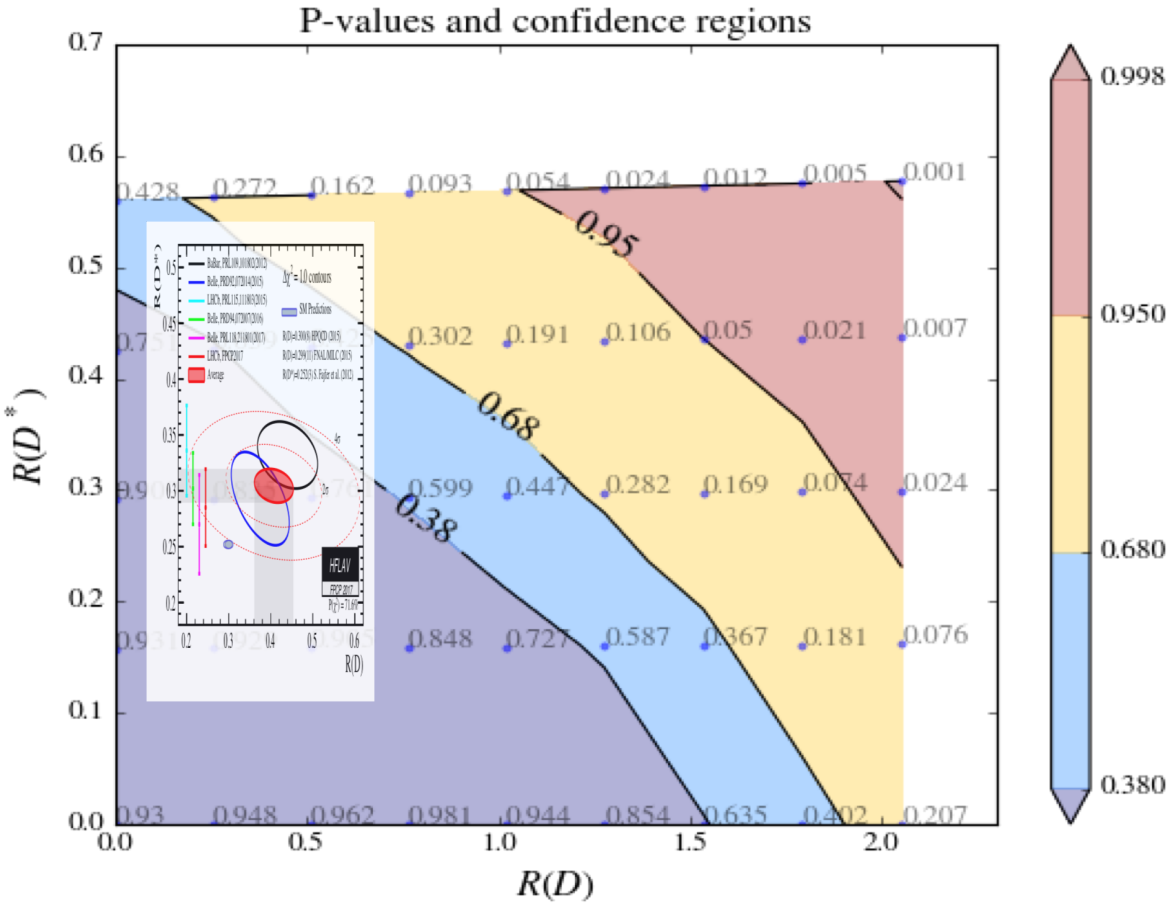


Figure 12.5: Confidence regions of $\mathcal{R}(D^{(*)})$ with world-average and SM prediction overlaid.

Chapter 13

Conclusions

We performed the first measurement of $\mathcal{R}(D^{(*)})$ using semileptonic tags and τ decays to hadrons and quote 68% and 95% confidence intervals. Unfortunately, we do not observe enough evidence to distinguish the Standard Model predictions from the current deviations in the world average.

Nevertheless, we have demonstrated that the low signal to background ratio in the τ to hadron channel poses its greatest difficulty not in statistical noise, but in systematic variations. Therefore, it is perhaps worth revisiting this channel only when systematic variations get suppressed; unfortunately, this suppression can only happen gradually over time as the community accumulates more knowledge about the underlying physical mechanisms.

We also learned that our analysis consumes characteristics of the simulated data that very poorly reflect those in the detector. Even though this was expected of the hadronic τ decays, one could argue that future iterations of this analysis could use more care in deciding which aspects of the simulated data to be used.

Despite the difficulties in obtaining a sensitive physics result, we have created a novel analysis toolchain that is highly robust and reusable. It has already been adopted by several child analyses, and has been running reliably for the last 2 years and counting. The primary value add of this toolchain are the highly performant statistical algorithms that were implemented; they were the reason for the negligible statistcal uncertainties. The additional value add, which is arguably more important, is that this toolchain has automated and offloaded the tedious and time consuming bookkeeping that analysts typically had to do, freeing them up to focus on the physics problem at hand.

Appendices

Appendix A

Graph Matching

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be directed graphs representing, respectively, true particle decays and reconstructed particle decays. The graphs are defined such that a vertex is inserted for each particle participating in the decay, and an edge (u, v) is inserted if v is a daughter particle of u.

Each vertex contains the following attributes:

A1 *id*: particle species identifier.

A2 *hit*: detector deposit pattern. Non-null only for vertices with no out-edges.

Definition 3 Let $u \in V_H$ and $u' \in V_H$. We say that u and u' are *semantically equivalent* if $u.id = u'.id$ and $u.hit = u'.hit$. Denote this with $u \sim u'$. (Note: we take the convention that NULL compares equal only to NULL.)

G and H further satisfies following properties:

P1 A set of vertices in G or H can have at most 1 common mother.

P2 Let $u \in V_G$ be a vertex with no out edges. There is at most one $u' \in V_H$ such that $u \sim u'$.

P3 G has exactly one vertex with no in-edges. We refer to this vertex as the *root* of G.

Definition 4 The subtree rooted at $u \in V_G$ is the subgraph induced by all vertices that are reachable from u. That is, it is the subgraph $G' = (V', E')$ with $V' = \{v \in V_G |\ v \text{ reachable from } u.\ \}$ and $E' = \{(u, v) \in E | u, v \in V'\}$.

Define the subtree rooted at $u' \in V_H$ similarly.

Let u be a vertex in G or in H. Denote $D(u)$ to be the set of all neighbors of u. By construction, $D(u)$ is the set that contains only vertices corresponding to daughters particles of u. We can now define, recursively, what it means for two decays trees from G and H to be "the same".

Definition 5 The subtree rooted at $u \in V_G$ is isomorphic to the subtree rooted at $u' \in V_H$ if the following hold:

1. $u \sim u'$.
2. There exists a bijection $f : D(u) \to D(u')$ such that for any $v \in D(u)$, v is tree-isomorphic to $f(v)$.

We sometimes will simply say that u is *tree-isomorphic* to u' and denote it as $u \cong u'$.

Proposition 1 *Vertices in G with no out-edges are tree-isomorphic to at most one vertex in H, and vice versa.*

Proof: Immediate by property **P2** and definiton 5. ■

Lemma 1 *Let u be any vertex in G. There is at most 1 vertex in $u' \in H$ such that $u \cong u'$.*

Proof: By induction on the height of the subtree rooted at u.

- Base case: true by proposition 1.
- Inductive case: Suppose u is tree-isomorphic to at least two distinct vertices in V_H; pick any two such vertices $s', t' \in V_H$. Since all subtrees rooted at vertices in $D(u)$ have lesser height, they are each tree-isomorphic to exactly 1 vertex in V_H. Denote this set of vertices as S. It follows that u must be tree-isomorphic to a common mother of S; however, property **P1** states that there can be at most 1 such mother. Contradiction.

■

Lemma 2 *Let $u' \in V_H$ and suppose each of its daughters are tree-isomorphic to some vertex in V_G; denote the set of these vertices as $S \subseteq V_G$. If u' is tree-isomorphic to any vertex in V_G, it must be that $u' \cong m$ where m is the common mother of S.*

Proof: Immediate by applying definition 5 and property **P1**. ■

Lemma 1 allows us to formally state the *particle graph matching problem*:

- Input: Graphs G and H.
- Output: Decide if the root vertex in G is tree-isomorphic to any vertex in H. If so, return it.

Theorem 3 *Algorithm 1 solves the particle graph matching problem correctly and runs in time $O(V + E)$.*

Proof: Algorithm 1 actually decides, for *every* vertex $u' \in V_H$, whether it is tree-isomorphic to any vertex in V_G, and if so finds it. It is correct by lemma 2 because it does so in depth first order. ■

Appendix B

Consistency Test

In order to validate the analysis procedure, especially the home-brewed software like the kernel density estimation, we perform a self-consistency test.

Suppose we have fixed component densities for the five event types and a set of proportions. While the exact values of the proportions and bandwidths used for the KDE's are not important, we use the proportions expected from SM prediction, $\vec{p} = (0.008, 0.0129, 0.125, 0.444, 0.410)$.

We can generate test datasets (of size $N = 8.7$ million) as follows:

1. Draw $x \sim unif(0, 1)$.
2. Determine the event type x correpsonds to based on $\vec{p}$, then generate a point from the corresponding event type KDE.
3. Repeat until we have N points.

This effectively removes all possible external discrepancies between the training data, used to construc the KDE's, and the test data, from which we attempt to extract $\vec{p}$.

We generate 300 such test datasets, and estimate the signal proportions for all datasets. Figure B.1 shows the distribution of difference between expected and extracted signal proportions, and we observe the clear unbiasedness of our estimation of the signal proportions.

This test also demonstrates the validity of using the bootstrap to estimate the variance of the extracted signal proportions: The standard deviation estimated based on the sample variance of the 300 test datasets agree well with the bootstrap estimation of the standard deviation (B.1).

	$D\tau$	$D^*\tau$
Bootstrap S.D.	0.000895	0.000812
Sample S.D.	0.000887	0.000826

Table B.1: Validation of bootstrap estimation of the variance by comparison to the sample variance of the results of the 300 simulated datasets.

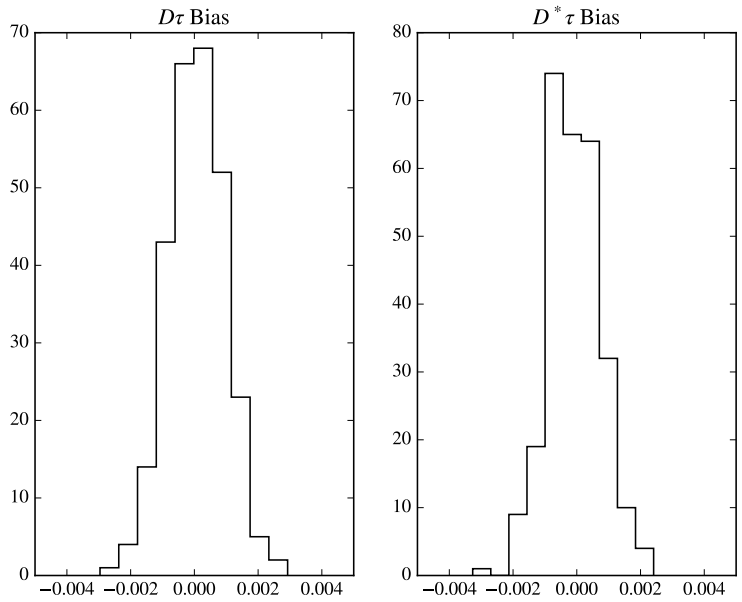
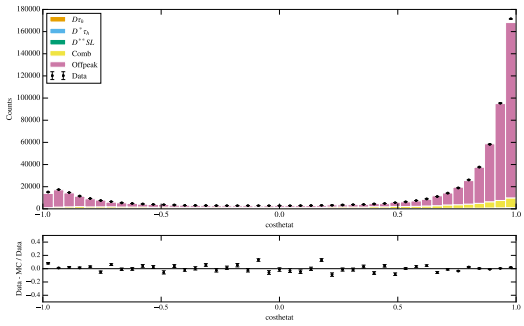


Figure B.1: Runtime benchmark. N is the number of training and query points and *log* is in base 2. $1M$ is approximately when $\log N = 20$

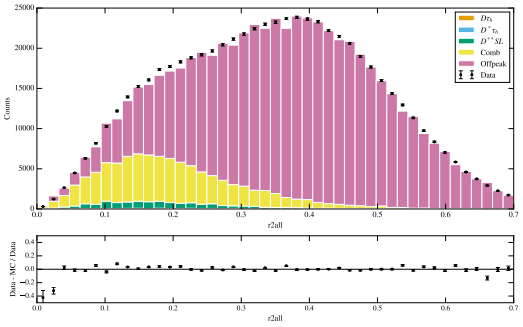
Appendix C

Sideband comparisons

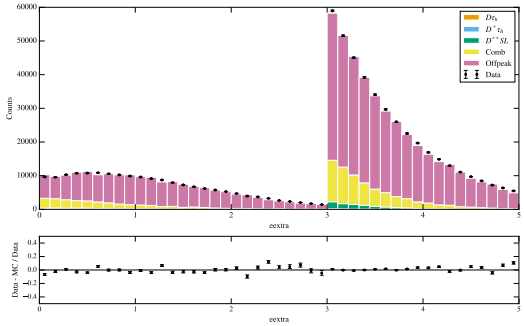
We show data and simulation comparisons in the sideband. This is an extension of the discussion in section 11.4.



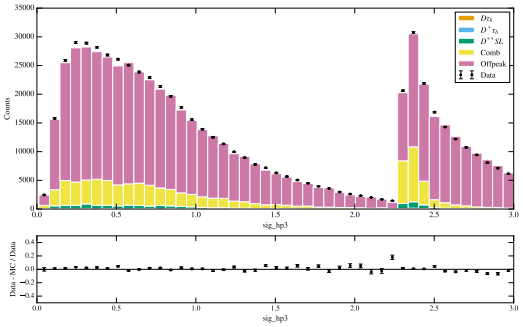
(a) $\cos\theta_T$.



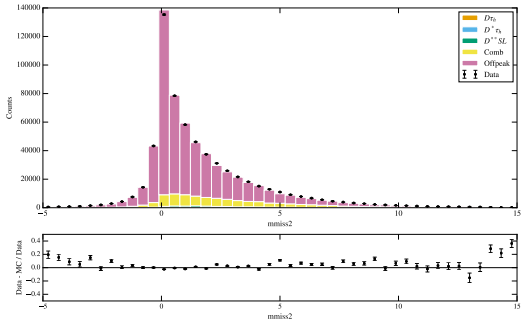
(b) R_2.



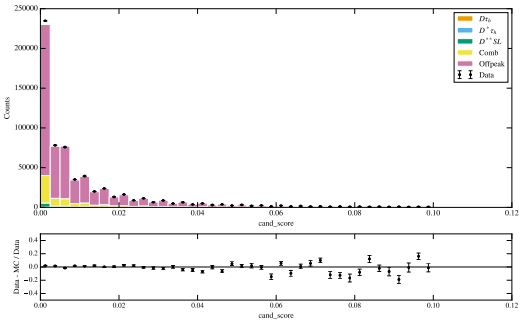
(c) E_{extra}. Left of the cliff belongs to the sideband sample where $|\vec{p}_h^{sig}| > 2.3$ GeV, while the right of the cliff belong to those with $E_{extra} > 3$ GeV.



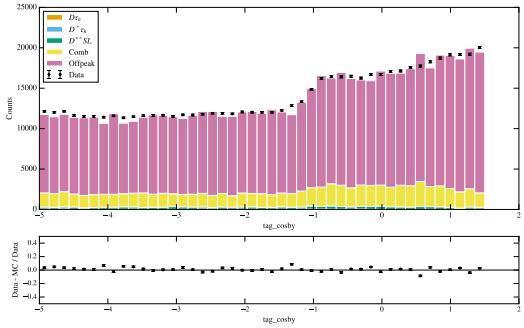
(d) $|\vec{p}_h^{sig}|$. Similar comments to those for E_{eextra} apply.



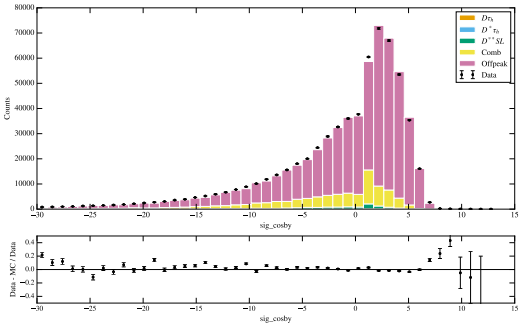
(e) M^2_{miss}.



(f) Candidate score.

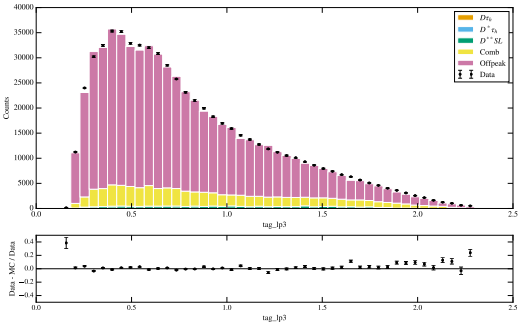


(g) $\cos\theta^{tag}_{BY}$.

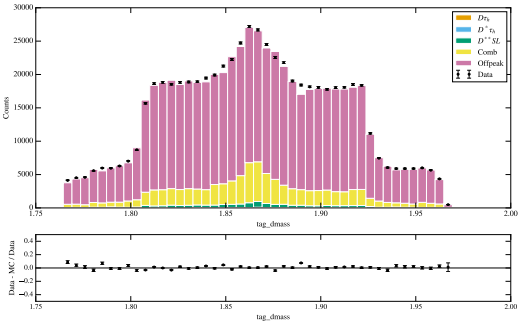


(h) $\cos\theta^{sig}_{BY}$.

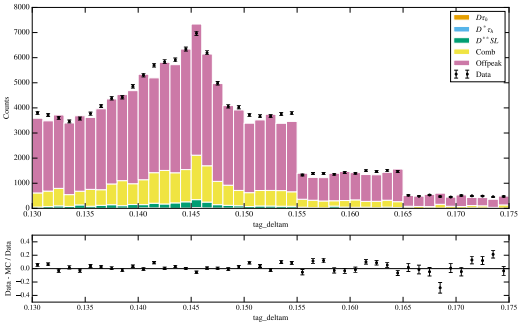
Figure C.1: Comparisons between data and MC for each event type in the sideband.



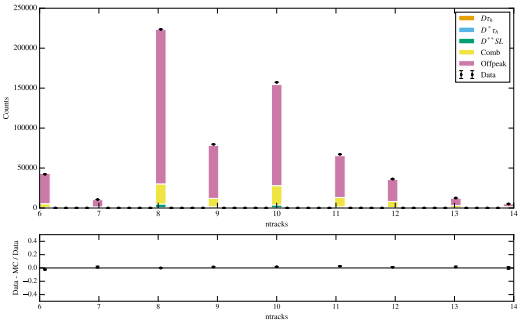
(a) $|\vec{p}_\ell^{\,tag}|$.



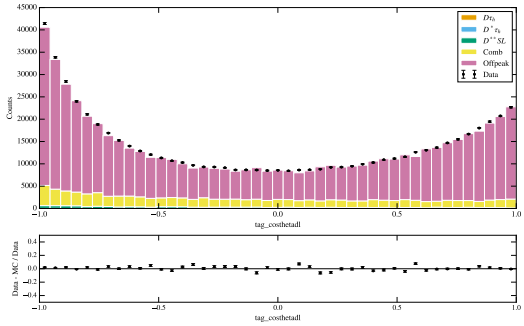
(b) m_D^{tag}.



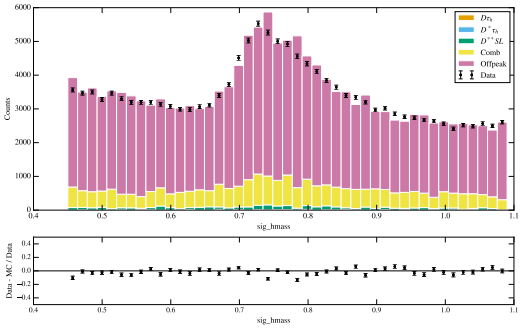
(c) Δm^{tag}.



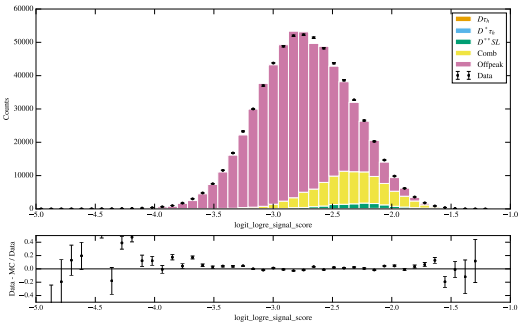
(d) N tracks.



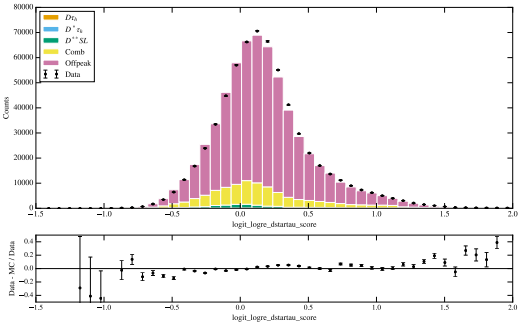
(e) $\cos\theta_{D\ell}$.



(f) m_h^{sig}.



(g) Z_1.



(h) Z_2.

Figure C.2: Comparisons between data and MC for each event type in the sideband.

Bibliography

[1] *BABAR* Collaboration, J. P. Lees *et al.*, Phys. Rev. D **88**, 072012 (2013).

[2] Belle Collaboration, M. Huschle, *et al.*, Phys. Rev. D **92**, 072014 (2015).

[3] Belle Collaboration, A. Abdesselam, *et al.*, (2016), arXiv:1603.06711 [hep-ex].

[4] LHCb Collaboration, R. Aaij *et al.*, Phys. Rev. Lett. **115**, 111803 (2015).

[5] Belle Collaboration, S. Hirose, Phys. Rev. Lett. **118**, 211801 (2017).

[6] HPQCD Collaboration, H. Na, C. M. Bouchard, G. P. Lepage, C. Monahan, and J. Shigemitsu, Phys. Rev. D **92**, 054510 (2015).

[7] S. Fajfer, J. F. Kamenik, and I. Nišandžić, Phys. Rev. D **85**, 094025 (2012).

[8] K. Hagiwara, A. D. Martin, and M. F. Wade, Nucl. Phys. **B327**, 569 (1989).

[9] J. G. Körner and G. A. Schuler, Zeitschrift für Physik C Particles and Fields **46**, 93 (1990).

[10] I. Caprini, L. Lellouch, and M. Neubert, Nucl. Phys. **B530**, 153 (1998), hep-ph/9712417.

[11] Heavy Flavor Averaging Group, Y. Amhis *et al.*, (2014), arXiv:1412.7515 [hep-ex].

[12] The Belle Collaboration, W. Dungle *et al.*, Phys. Rev. D **82**, 112007 (2010).

[13] *BABAR* Collaboration, B. Aubert *et al.*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **729**, 615 (2013).

[14] *BABAR* Collaboration, B. Aubert *et al.*, Phys. Rev. Lett. **100**, 231803 (2008).

[15] *BABAR* Collaboration, B. Aubert *et al.*, Phys. Rev. D **79**, 012002 (2009).

[16] *BABAR* Collaboration, B. Aubert *et al.*, Phys. Rev. D **77**, 032002 (2008).

[17] *BABAR* Collaboration, B. Aubert *et al.*, Phys. Rev. Lett. **100**, 151802 (2008).

[18] Particle Data Group, K. A. Olive *et al.*, Chin. Phys. **C38**, 090001 (2014).

[19] M. S. Andersen, J. Dahl, and L. Vandenberghe, CVXOPT: A Python package for convex optimization, http://cvxopt.org/index.html (2015).

[20] B. W. Silverman, Chapman & Hall, London (1986).

[21] R. Brun and F. Rademakers, Nuclear Instruments and Methods in Physics Research Section A **389**, 81 (1996).

[22] A. G. Gray and A. W. Moore, Proceedings of the third SIAM international conference on data mining. (2003), http://dx.doi.org/10.1137/1.9781611972733.19.

[23] NVIDIA coorporation, H. Nguyen, Upper Saddle River : Addison-Wesley (2007).

[24] J. Barnes and P. Hut, Nature **324** (1986).

[25] S. M. Kay, Prentice Hall (1993).

[26] L. Breiman, Mach. Learn. **45**, 5 (2001).

[27] F. Pedregosa *et al.*, Journal of Machine Learning Research **12**, 2825 (2011).

[28] Y. Abu-Mostafa, M. Magdon-Ismail, and H. T. Lin, http://amlbook.com/ (2012).

[29] K. P. Murphy, MIT press (2012).

[30] S. Boyd and V. Vandenberghe, Cambridge University Press (2004).

[31] C. J. Stone, Ann. Statist. **12**, 1285 (1984).

[32] A. Gray and A. Moore, Advances in Neural Information Processing Systems 13 , 521 (2000).

[33] J. Bentley, Commun. ACM **18**, 509 (1975).

[34] D. Scora and N. Isgur, Phys. Rev. D **52**, 2783 (1995).

[35] CLEO Collaboration, J. E. Duboscq *et al.*, Phys. Rev. Lett. **76**, 3898 (1996).

[36] A. K. Leibovich, Z. Ligeti, I. W. Stewart, and M. B. Wise, Phys. Rev. D **57**, 308 (1998).

[37] F. U. Bernlochner and Z. Ligeti, (2016), arXiv:1606.09300.

[38] F. U. Bernlochner, Z. Ligeti, and S. Turczyk, Phys. Rev. D **85**, 094033 (2012).